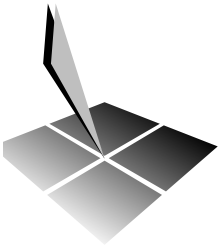


FieldPoint™

FieldPoint™ FP-3000 Network Module User Manual

NATIONAL INSTRUMENTS™

January 1999 Edition
Part Number 322169A-01

Worldwide Technical Support and Product Information

http://www.natinst.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, Denmark 45 76 26 00, Finland 09 725 725 11,
France 0 1 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886,
Spain (Madrid) 91 640 0085, Spain (Barcelona) 93 582 0251, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix of this manual.

© Copyright 1999 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The FieldPoint FP-3000 network module is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

FieldPoint™, Lookout™, `natinst.com`™, and NI-FBUS™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

Contents

About This Manual

Conventions xv
Related Documentation xv

Chapter 1
FP-3000 Network Module Overview

Overview of the FP-3000 Network Module 1-1
Features of the FP-3000 Network Module 1-3
Function Blocks 1-3
PID Control 1-3
Block Instantiation 1-3
Interoperability 1-4
Link Active Scheduler (LAS) Functionality 1-4
HotPnP (Hot Plug and Play) 1-4
Field Upgradability 1-4

Chapter 2
Installation and Configuration

Install the Device Description File 2-1
Updating the Device Description 2-3
Mount the FP-3000 and Terminal Bases 2-4
Mounting the FP-3000 on a DIN Rail 2-4
Connecting Terminal Bases with DIN Rail Mounting 2-5
Removing the FP-3000 from the DIN Rail 2-6
Mounting the FP-3000 to a Panel 2-6
Connecting Terminal Bases with Panel Mounting 2-7
Removing the FP-3000 and Terminal Bases from the Panel 2-8
Mount I/O Modules onto Terminal Bases 2-8
Connect Power to the FP-3000 2-9
Calculating Power for a FieldPoint Bank 2-9
Power-On Self Test (POST) 2-10
Connect the FP-3000 to the Fieldbus Network 2-10
LED Indicators 2-12
HotPnP (Hot Plug and Play) 2-14
HotPnP During Operation 2-14
Inserting New I/O Modules 2-15
Replacing I/O Modules 2-15
Updating the FP-3000 Firmware 2-15

Chapter 3
Example Applications

Initial Power On: Assigning Address and Device Tag 3-1
Example 1: Converting a 4–20 mA Pressure Sensor to Fieldbus Using FP-3000 3-2
Getting Started 3-2
Convert the Pressure Sensor Reading 3-2
Create Function Block 3-2
Assign a Tag to the New Block 3-3
Select the Module and Channel 3-3
Set the Input Range 3-4
Scale the Reading 3-4
Set Up Scheduling 3-5
Bring the Block Online 3-6
Example 2: Temperature Control with the FP-3000 3-6
Getting Started 3-6
Taking Temperature Readings 3-7
Create an FP-TC-120 Block 3-7
Assign a Tag to the New Block 3-7
Select the Module and Channel 3-7
Set the Input Range and Thermocouple Type 3-8
Scale the Reading 3-8
Set Up Scheduling 3-9
Bring the Block Online 3-10
Controlling a Heating Element 3-10
Instantiate an FP-AO-200 Block 3-10
Assign a Tag to the New Block 3-10
Select the Module and Channel 3-11
Set the Output Range 3-11
Scale the Output 3-11
Set Up Scheduling 3-12
Bring the Block Online 3-12
PID Control 3-13
Instantiate a PID Block 3-13
Assign a Tag to the New Block 3-13
Scale the PID 3-13
Connect the PID to the AI and AO Blocks 3-14
Download and Bring the Loop into Auto 3-14
Tune the PID 3-15
Alarming 3-15

Chapter 4
Block Reference

Block Overview 4-1
Function Blocks 4-1
Resource Block 4-2
Types of Function Blocks 4-2
AI (Analog Input) 4-2
AO (Analog Output) 4-2
PID (Proportional–Integral–Derivative) 4-3
DI (Discrete Input) 4-3
DO (Discrete Output) 4-3
CDO (Complex Discrete Output) 4-3
LOG (FieldPoint Log Block) (FP-3000 Specific) 4-4
STAT (FieldPoint Statistics Block) (FP-3000 Specific) 4-4
Function Blocks and FieldPoint Modules 4-4
PID Control 4-5
Alarming 4-6
Alarm Parameters 4-6
UNACKNOWLEDGED 4-7
ALARM_STATE/UPDATE_STATE 4-7
TIME_STAMP 4-8
SUBCODE 4-8
VALUE 4-8
Status and Mode Handling Overview 4-8
Status Handling 4-8
Quality 4-9
Substatus 4-9
Limit 4-10
MODE_BLK Parameter and Mode Handling 4-10
Target Mode 4-10
Actual Mode 4-11
Permitted Mode 4-12
Normal Mode 4-12
FP-3000 Specific Parameters 4-13
CFG_OPTS 4-13
DEV_OPTS 4-13
EXECUTION_STATISTICS 4-13
FIELDPOINT_CHANNEL 4-14
FIELDPOINT_MODULE 4-14
FP_AI_100_RANGE 4-14
FP_AI_110_RANGE 4-14
FP_AI_111_RANGE 4-14
FP_AO_200_RANGE 4-15

FP_CJC_SOURCE 4-15
FP_MOD_STATUS 4-15
FP_NOISE_REJECTION 4-15
FP_PWM_520_PERIOD 4-16
FP_RTD_122_RANGE 4-16
FP_RTD_TYPE 4-16
FP_TC_120_RANGE 4-16
FP_TC_120_CJ_RANGE 4-16
FP_THERMOCOUPLE_TYPE 4-16
LAST_BLOCK_EVENT 4-16
VERSION_INFORMATION 4-17

Appendix A
Configuring the FP-3000

Simulate Enable A-1
Write Lock A-2
Reset A-2

Appendix B
Troubleshooting

Fieldbus Communication Problems B-1
Setting Device Tag and Network Address B-1
I/O Module Problems B-3
Software Configuration Problems B-3

Appendix C
Fieldbus Parameters

ACK_OPTION C-1
ALARM_HYS C-1
ALARM_SUM C-1
ALERT_KEY C-1
ALG_RUN_TIME C-1
BAL_TIME C-1
BINARY_CL C-1
BINARY_OP C-2
BKCAL_HYS C-2
BKCAL_IN C-2
BKCAL_OUT C-2
BLOCK_ALM C-2
BLOCK_ERR C-2
BLOCK_RESET C-4

BYPASS C-4
CHANNEL C-4
CHECKBACK C-4
CLR_FSTATE C-5
CONFIRM_TIME C-5
CONTROL_OPTS C-6
CYCLE_SEL/CYCLE_TYPE C-6
DD_RESOURCE C-6
DD_REV C-7
DEV_REV C-7
DEV_TYPE C-7
DV_HI_ALM C-7
DV_HI_LIM C-7
DV_HI_PRI C-7
DV_LO_ALM C-7
DV_LO_LIM C-7
DV_LO_PRI C-7
FAULT_STATE C-7
FEATURE_SEL/FEATURES C-8
FF_GAIN C-8
FF_SCALE C-8
FF_VAL C-9
FIELD_VAL C-9
FP_AUTOCONFIGURE C-9
FP_MOD_LIST C-9
FREE_SPACE C-9
FREE_TIME C-9
GAIN C-9
GRANT_DENY C-9
HARD_TYPES C-10
HI_ALM C-10
HI_HI_ALM C-10
HI_HI_LIM C-10
HI_HI_PRI C-10
HI_LIM C-10
HI_PRI C-10
IO_OPTS C-10
L_TYPE C-11
LIM_NOTIFY C-12
LO_ALM C-12
LO_LIM C-12
LO_LO_ALM C-12
LO_LO_LIM C-12
LO_LO_PRI C-12

LO_PRI C-13
LOW_CUT C-13
MANUFAC_ID C-13
MAX_NOTIFY C-13
MEMORY_SIZE C-13
MIN_CYCLE_T C-13
MODE_BLK C-13
NV_CYCLE_T C-14
OP_CMD_CXO C-15
OUT C-15
OUT_HI_LIM C-15
OUT_LO_LIM C-15
OUT_SCALE C-15
PV C-16
PV_FTIME C-16
PV_SCALE C-16
RATE C-16
RCAS_IN C-16
RCAS_OUT C-17
RESET C-17
RESTART C-17
ROUT_IN C-17
ROUT_OUT C-17
RS_STATE C-18
SAFEGUARD_CL C-18
SAFEGUARD_OP C-18
SET_FSTATE C-18
SHED_OPT C-18
SHED_RCAS C-19
SHED_ROUT C-19
SIMULATE C-19
SP_HI_LIM C-19
SP_LO_LIM C-19
SP_RATE_DN C-20
SP_RATE_UP C-20
ST_REV C-20
STATUS_OPTS C-20
STRATEGY C-21
TAG_DESC C-21
TEST_RW C-21
TRK_IN_D C-21
TRK_SCALE C-21
TRK_VAL C-21
UPDATE_EVT C-22

WRITE_ALM....C-22
WRITE_LOCK....C-22
WRITE_PRI....C-22
XD_SCALE....C-22

Appendix D
Advanced Function Block Behavior

Cascade Initialization....D-1
Parameter Connections for Cascade Initialization....D-1
Mode and Status Behavior during Cascade Initialization....D-2
Remote Cascades....D-3
Bypassing Cascade Initialization....D-4
Fault State and Mode Shedding....D-4
Fault State....D-4
Mode Shedding....D-4

Appendix E
Specifications

Appendix F
Technical Support Resources

Glossary

Index

Figures

Figure 1-1. Fieldbus Network Setup....1-2

Figure 2-1. NI-FBUS Configuration Utility Dialog Box....2-2
Figure 2-2. DD Info Dialog Box....2-2
Figure 2-3. Import DD Dialog Box....2-3
Figure 2-4. fbconf Dialog Box....2-3
Figure 2-5. DIN Rail Clip....2-4
Figure 2-6. Mounting the FP-3000 onto a DIN Rail....2-5
Figure 2-7. Connecting Terminal Bases....2-6
Figure 2-8. Installing the Network Panel Mount Accessory....2-7
Figure 2-9. Mounting I/O Module to Terminal Base....2-8
Figure 2-10. FP-3000 Power Connector Pinout....2-9
Figure 2-11. Fieldbus Connectors on the FP-3000....2-11

Figure 2-12. FP-3000 Connector Pinout 2-12
Figure 2-13. LEDs on the FP-3000 2-12
Figure 2-14. FP-3000 Firmware Update Dialog Box 2-16
Figure 2-15. FP-3000 Search Dialog Box 2-17
Figure 2-16. Select FP-3000 Module Dialog Box 2-18

Figure 3-1. Setting the Channel Dialog Box 3-3
Figure 3-2. Downloading Configuration Dialog Box 3-5
Figure 3-3. Set the Input Range and Thermocouple Type Dialog Box 3-8
Figure 3-4. PID Block Connections Dialog Box 3-14
Figure 3-5. High Limit Alarm Parameters Dialog Box 3-16
Figure 3-6. PID Alarm Connection Dialog Box 3-17

Figure 4-1. PID Function Block Application Dialog Box 4-6

Figure A-1. Configuration Toggle Switches A-1

Figure D-1. Parameter Connections for Cascade Initialization D-2
Figure D-2. Remote Cascade Model D-3

Tables

Table 2-1. Description of Fieldbus NETWORK LED States 2-13
Table 2-2. STATUS LED Flashes and Corresponding Error Conditions 2-14

Table 4-1. CDO Block Interlock Priorities 4-3
Table 4-2. Function Blocks and FieldPoint Modules 4-5
Table 4-3. Quality Values 4-9
Table 4-4. Limit Values 4-10
Table 4-5. Target Modes 4-10
Table 4-6. Actual Modes 4-11
Table 4-7. Configuration Options 4-13
Table 4-8. Device Options 4-13
Table 4-9. Execution Statistics 4-14
Table 4-10. Module Status 4-15
Table 4-11. Block Events 4-17

Table B-1. Fieldbus Communication Problems B-2
Table B-2. I/O Module Problems B-3
Table B-3. Generic Software Configuration Problems B-4
Table B-4. Resource Block Configuration Problems B-6

Table C-1. Error Codes C-3
Table C-2. Block Reset Options C-4

Table C-3. Checkback States C-5
Table C-4. Control Options C-6
Table C-5. Feature Parameter Options C-8
Table C-6. Hard Types C-10
Table C-7. Operation Bitmasks C-11
Table C-8. Linearization Types C-12
Table C-9. Operational Modes C-14
Table C-10. Command Parameters C-15
Table C-11. OUT_SCALE Parameter C-16
Table C-12. Restart Values C-17
Table C-13. Device States C-18
Table C-14. Shed Conditions C-19
Table C-15. Status Options C-20
Table C-16. Scaling Parameter Values C-22

Table D-1. Mode Shedding Options D-5

About This Manual

This manual describes how to use your FieldPoint FP-3000 Network Module.

Conventions

The following conventions appear in this manual:

»	The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a note, which alerts you to important information.
	This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.
bold	Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- Operating Instructions (for network module, terminal bases, and I/O modules)
- Fieldbus Foundation's *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium Application Guide*
- *Fieldbus Standard for Use in Industrial Control Systems, Part 2,* ISA-550.01.1992

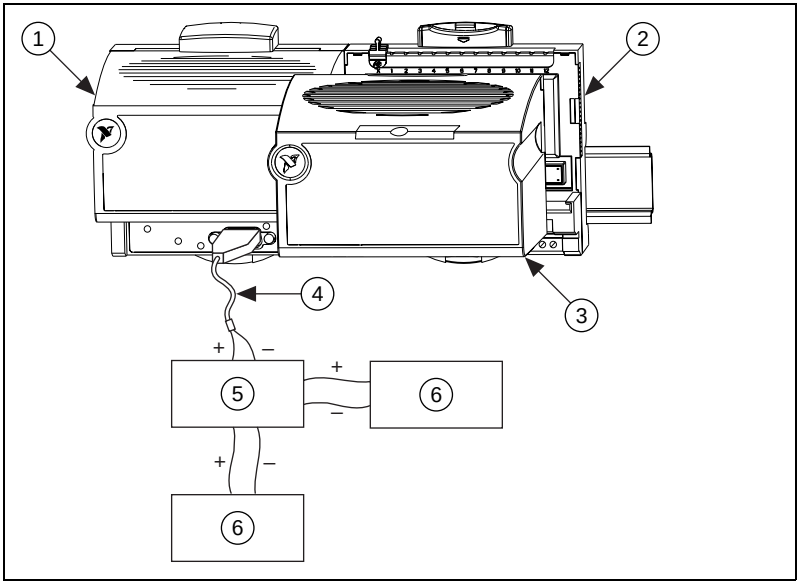
1

FP-3000 Network Module Overview

This chapter provides an overview of the FP-3000 network module.

Overview of the FP-3000 Network Module

The FP-3000 is an intelligent network interface and controller module that manages a bank of up to nine FieldPoint I/O modules and terminal bases. The FP-3000 network module and the terminal bases snap together to form a high-speed data bus for communications between the FP-3000 network module and any I/O modules in the bank. The FP-3000 includes an H1 Fieldbus interface for direct connection to an H1 FOUNDATION Fieldbus segment. A Fieldbus segment is a multidrop network and can consist of up to 32 H1-compliant devices. You can connect a Fieldbus segment to up to 32 FP-3000 network modules. Figure 1-1 shows an FP-3000 connected to a Fieldbus network.



1 FP-3000 Network Module 2 Terminal Base	3 I/O Module 4 Fieldbus Trunk	5 Fieldbus 6 Fieldbus Devices

Figure 1-1. Fieldbus Network Setup

The FP-3000 network module provides a Fieldbus interface to conventional analog and discrete I/O devices. For example, the FP-3000 makes a 4–20 mA pressure transmitter connected to a FieldPoint 8-channel analog input module behave like a Fieldbus pressure transmitter. By using an FP-3000 network module, you can significantly reduce wiring and installation costs. Instead of running a pair of wires from each 4–20 mA device to your controller, you can mount an FP-3000 network module in the field and run a single pair of wires (called the trunk) from your PC to the FP-3000. You connect the 4–20 mA devices to the FieldPoint I/O modules by short stretches of wire.

Features of the FP-3000 Network Module

Function Blocks

Conventional devices connected to I/O modules are made visible as Fieldbus function blocks. Function blocks are software modules which describe the fundamental elements of an I/O or control system. The FP-3000, like any FOUNDATION Fieldbus–compliant device, has one or more function blocks. The function blocks in different devices can be connected to form a distributed control system.

The FP-3000 implements FOUNDATION Fieldbus–compliant I/O function blocks, such as Analog Input (AI), Analog Output (AO), Discrete Input (DI), and Discrete Output (DO). These blocks provide functionality such as scaling, trending, and alarming. For example, when you connect a 4–20 mA pressure transmitter to a FieldPoint I/O, you can configure an FP-3000 Analog Input function block to convert from 4–20 mA to engineering units. You can set up alarm limits so that the FP-3000 sends an alarm when the pressure exceeds the limits. The FP-3000 network module can also collect trend samples and broadcast them to applications on a PC.

PID Control

The FP-3000 implements the FOUNDATION Fieldbus–compliant PID control function block. This PID can be used to control either an analog output element connected to FieldPoint I/O or a native Fieldbus device, such as a valve, connected to the Fieldbus network. The FP-3000 executes the PID and other function blocks deterministically in accordance to a configured schedule.

Block Instantiation

You can instantiate (create multiple copies of) the PID function block on an as-needed basis. For example, if you are adding a new loop to an existing Fieldbus network, you can instantiate a PID function block in the FP-3000 to control the loop. You can also instantiate the I/O function blocks on an as-needed basis. If you have an 8-channel Analog Input module and you are using only two channels, you would instantiate two AI function blocks. You can instantiate additional AI function blocks when you use additional channels.

Interoperability

The FP-3000 network module can send or receive data from any Fieldbus-compliant device. The PID block in the FP-3000 can get its input from any Fieldbus-compliant device; it can also control any Fieldbus-compliant output device.

The control and I/O functionality of the FP-3000 can be configured by any Fieldbus configurator, such as the National Instruments NI-FBUS Configurator. This is possible because all of the features added by National Instruments are described using Device Descriptions. Any Fieldbus-compliant HMI package or OPC server can also access the FP-3000 function blocks.

Link Active Scheduler (LAS) Functionality

Fieldbus networks require a Link Active Scheduler (LAS) to control communications on the Fieldbus. The FP-3000 can act as a primary or back-up Link Active Scheduler. This enables the FP-3000 to execute PID and other function blocks without a PC connected to the network. If a PC is connected and the PC goes down, the FP-3000 takes over the bus and executes the control without causing a bump.

HotPnP (Hot Plug and Play)

FP-3000 network modules can be added or removed from H1 Fieldbus networks without affecting other Fieldbus devices.

You can insert or remove I/O modules into FieldPoint terminal bases while the power is on, even if the FP-3000 is already operational. The other I/O modules connected to the FP-3000 are fully operational during this process. You do not need to power down the FP-3000, Fieldbus network, or even a bank to insert or remove I/O modules. You do not need to restart the host computer software to use the HotPnP feature. You can replace an I/O module only with another I/O module of the same type.

Field Upgradability

You can download new versions of the firmware to the FP-3000 over the Fieldbus network without powering down the Fieldbus network or affecting the operation of other devices. You should use the National Instruments FP-3000 Firmware Update utility to download the firmware. This feature lets you take advantage of new function blocks as they are made available from National Instruments.

2

Installation and Configuration

This chapter describes how to install device description files, mount your FieldPoint FP-3000 network module, connect the terminal bases, connect power to the network module, and connect the FP-3000 to a Fieldbus network.

Install the Device Description File

The Device Description file (DD) contains a list of the types of function blocks and parameters supported by the FP-3000, along with online help describing the uses of given parameters. Before you can use the FP-3000 with NI-FBUS (or other host software), you must install the device description file (shipped with the device) on the host computer or computers. After the DD file for the FP-3000 network module is initially installed, the DD works for all FP-3000s on the Fieldbus connected to the computer on which the DD has been installed. To install the DD for use with NI-FBUS, complete the following steps:

Note This process is correct for use with National Instruments NI-FBUS. The process can vary with other host software packages.

1. Install your Fieldbus interface and NI-FBUS software, if you have not done so already.
2. Insert the Device Description diskette (shipped with the FP-3000) into the disk drive of the host computer.
3. Select **Start»Programs»NI-FBUS»Interface Config**. The following dialog box appears.

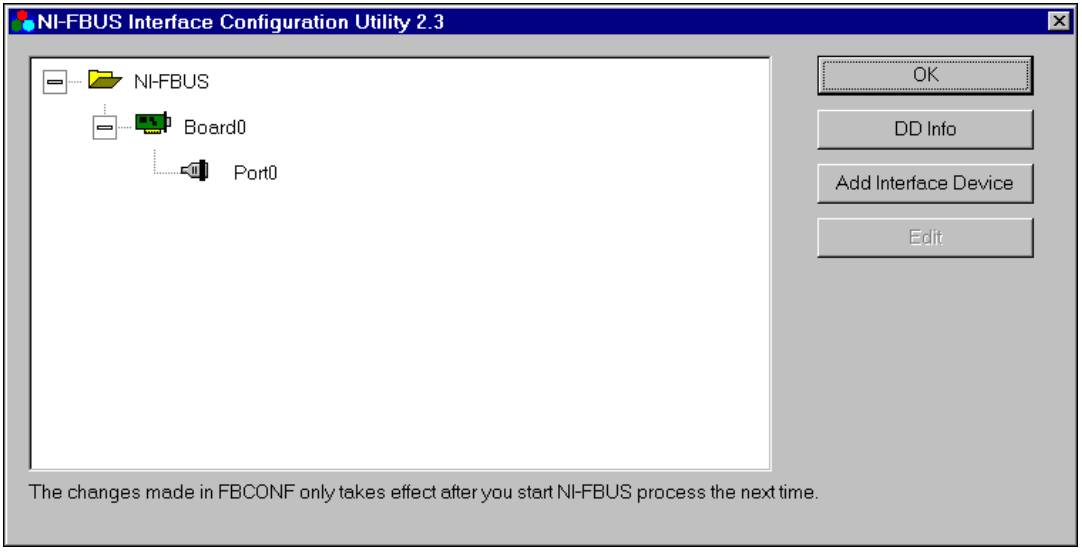


Figure 2-1. NI-FBUS Configuration Utility Dialog Box

4. Click on the **DD Info** button. The following dialog box appears.

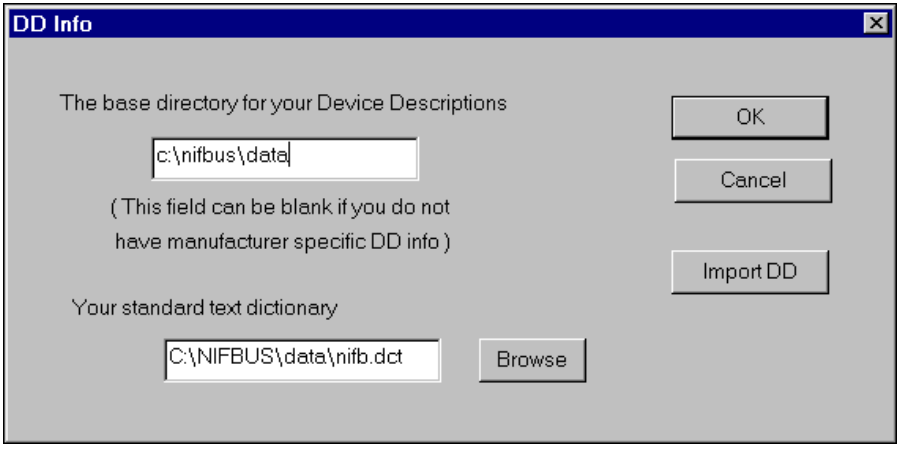


Figure 2-2. DD Info Dialog Box

5. Click on the **Import DD** button. The following dialog box appears.

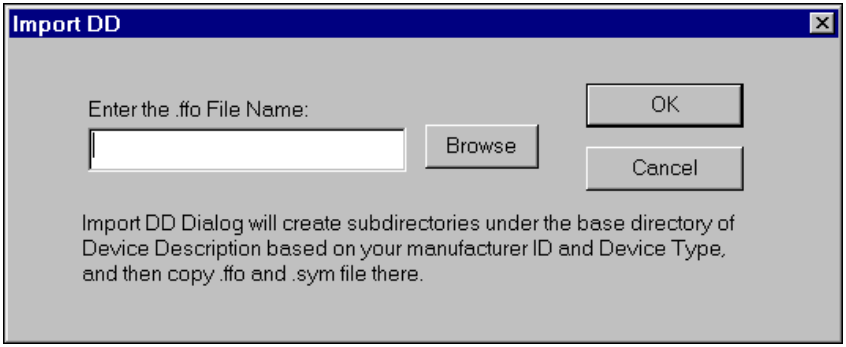

Import DD
Enter the .ffo File Name:
Browse
OK
Cancel
Import DD Dialog will create subdirectories under the base directory of Device Description based on your manufacturer ID and Device Type, and then copy .ffo and .sym file there.

Figure 2-3. Import DD Dialog Box

6. Enter the file name for the device description into the entry field, then click on **OK**. If the import process is successful, the following dialog box appears, indicating that the software installation is complete.

Figure 2-4. fbconf Dialog Box

You only need to install the DD file one time for a version of the firmware. You do not have to repeat the DD installation for each FP-3000 connected to your computer.

Updating the Device Description

Any enhancement to the FP-3000 functionality, such as the addition of new function blocks or support of new types of I/O modules, results in a new Device Description file describing the features of the FP-3000. You must install the new Device Description files. If you are using NI-FBUS Communications manager or the NI-FBUS Configuration, refer to the section *Updating the FP-3000 Firmware* for instructions on installing the new Device Description file.

Mount the FP-3000 and Terminal Bases

You can mount your FieldPoint system either to a DIN rail or directly on a panel. Panel mounting is generally the more secure option, but DIN rail mounting might be more convenient for your application. The following sections give instructions for both mounting methods.

Mounting the FP-3000 on a DIN Rail

The FP-3000 has a simple rail clip for reliable mounting onto a standard 35 mm DIN rail. Follow these steps to mount the FP-3000 on a DIN rail.

1. Use a flat-bladed screwdriver to open the DIN rail clip to the unlocked position, as shown in Figure 2-5.

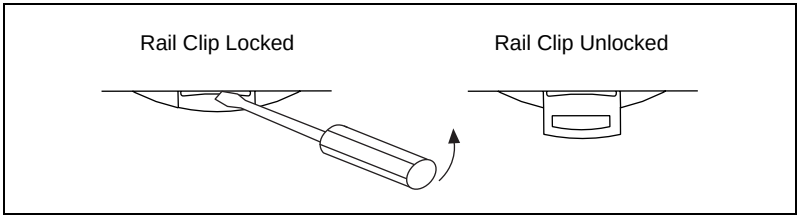


Figure 2-5. DIN Rail Clip

2. Hook the lip on the rear of the FP-3000 onto the top of a 35 mm DIN rail and press the FP-3000 down onto the DIN rail, as shown in Figure 2-6.

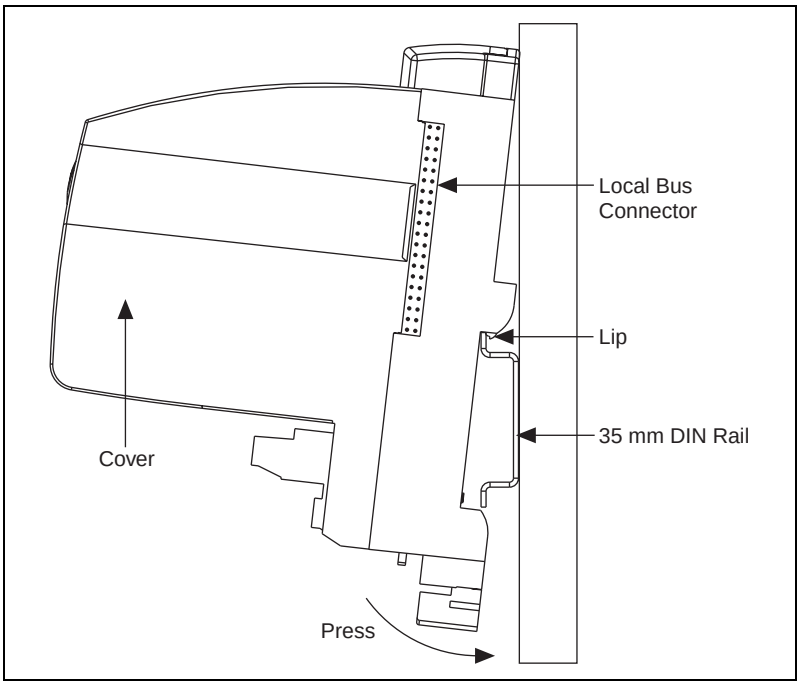


Figure 2-6. Mounting the FP-3000 onto a DIN Rail

3. Slide the FP-3000 to the desired position along the DIN rail. After the FP-3000 is in position, lock it to the DIN rail by pushing the rail clip to the locked position, as shown in Figure 2-5.

After the FP-3000 is mounted to the DIN rail, connect the terminal base to the FP-3000 as explained in the next section, *Connecting Terminal Bases with DIN Rail Mounting*.

Connecting Terminal Bases with DIN Rail Mounting

Follow these steps to connect a terminal base to an FP-3000 network module using DIN rail mounting.

Caution To avoid damaging the FP-3000 and the terminal bases, make sure that power is not applied to the FP-3000 while you install or remove terminal bases.

1. Mount the terminal base onto the DIN rail in the same way you installed the FP-3000.
2. Attach the terminal base to the FP-3000 by firmly mating the local bus connectors.

3. To add more terminal bases, install them on the rail and connect their local bus connectors together. A single FP-3000 can support up to nine terminal bases.
4. Place the protective cover (from the bag of accessories that came with your FP-3000) onto the local bus connector of the last terminal base on the bank, as shown in Figure 2-7.

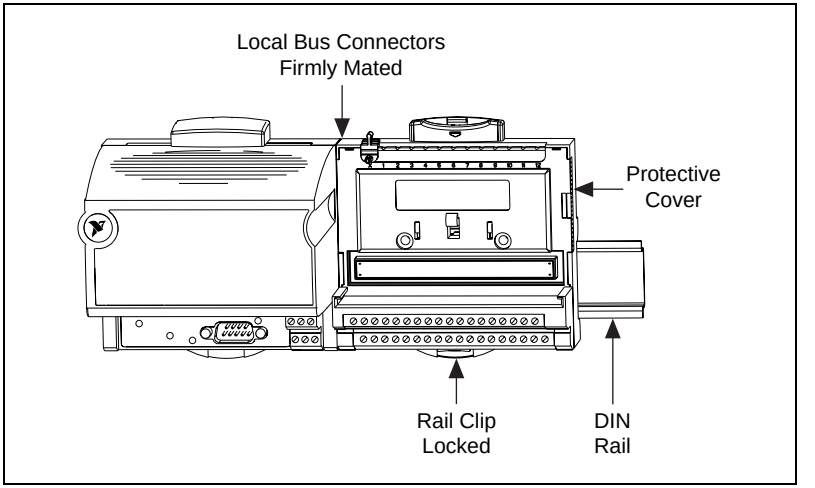


Figure 2-7. Connecting Terminal Bases

Removing the FP-3000 from the DIN Rail

To remove an FP-3000 network module, unlock it from the DIN rail by placing a screwdriver in the slot on the rail clip and opening it to the unlocked position, as shown in Figure 2-5. Then, disconnect the FP-3000 from the local bus connector of the terminal base, and lift the FP-3000 off the rail.

Mounting the FP-3000 to a Panel

Follow these steps to install the optional FieldPoint network panel mount accessory and mount the FP-3000 network module to a panel. You can order the panel mount accessory, part number 777609-01, from National Instruments.

1. Use a flat-bladed screwdriver to open the rail clip to the unlocked position, as shown in Figure 2-5.
2. Snap the panel mount accessory onto the module, as shown in Figure 2-8.

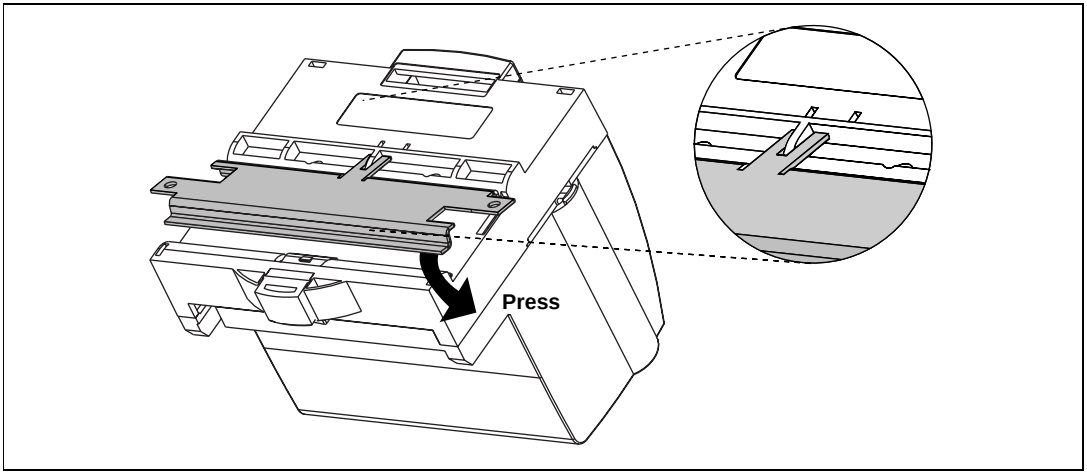


Figure 2-8. Installing the Network Panel Mount Accessory

3. Lock the panel mount accessory into place by pushing the rail clip to the locked position, as shown in Figure 2-5.
4. Mount the FP-3000 to your panel with the panel mount accessory. The installation guide that came with the panel mount accessory includes a guide that you can use to drill pilot holes for mounting the FP-3000.

Connecting Terminal Bases with Panel Mounting

You can install terminal bases directly, without using the panel mount accessory needed to mount the FP-3000 network module. Follow these steps to connect terminal bases to a network module using panel mounting.

Caution To avoid damaging the FP-3000 and the terminal bases, make sure that power is not applied to the FP-3000 while you install or remove terminal bases.

1. Drill pilot holes in the panel to mount the terminal bases. A drilling guide is provided with the network module panel mount accessory.
2. Attach the terminal base to the FP-3000 by firmly mating the local bus connectors.
3. Bolt, screw, or otherwise fasten the terminal base to the panel. Make sure that the local bus connectors remain firmly mated after the terminal base is mounted.
4. To add more terminal bases, repeat Steps 1 through 3, mating the local bus connectors of each new terminal base to the connector of the last installed base. If all the pilot holes were correctly drilled, the local bus

connectors should remain firmly mated after all the bases are mounted to the panel.

5. Place the protective cover (from the bag of accessories that came with your FP-3000) onto the local bus connector of the last terminal base on the bank.

Removing the FP-3000 and Terminal Bases from the Panel

To remove an FP-3000 network module and terminal bases from the panel, reverse the process described in the previous sections, *Mounting the FP-3000 to a Panel* and *Connecting Terminal Bases with Panel Mounting*. First remove the terminal bases, starting with the last one, then remove the network module.

Mount I/O Modules onto Terminal Bases

Follow these steps to connect an I/O module to a terminal base:

1. Position the first module with its alignment slots aligned with the guide rails on the terminal base, as shown in Figure 2-9.
2. Firmly press the module onto the terminal base. The terminal base latch locks the I/O module into place.
3. Repeat this procedure to install additional I/O modules onto terminal bases.

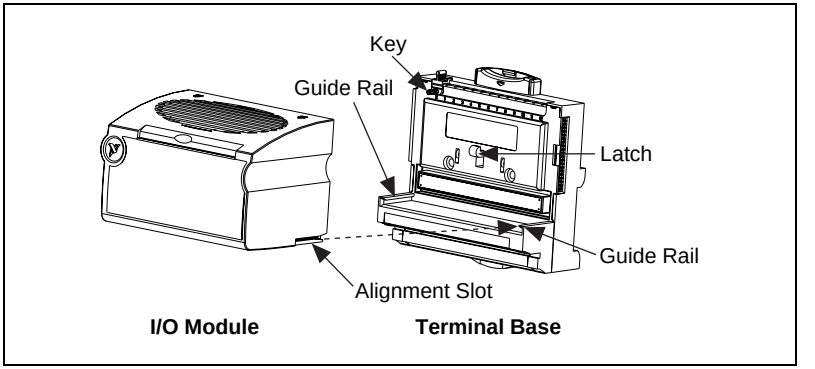


Figure 2-9. Mounting I/O Module to Terminal Base

Connect Power to the FP-3000

An 11–30 VDC power supply is required by each FP-3000 on your network. The FP-3000 filters and regulates this supplied power and provides power for all the I/O modules in the bank. Therefore, you do not need to provide power separately to each FieldPoint I/O module in the bank.

The power connector is a 6-pin screw terminal power connector whose pinout is shown in Figure 2-10. See Figure 2-11 for the location of the power connector.

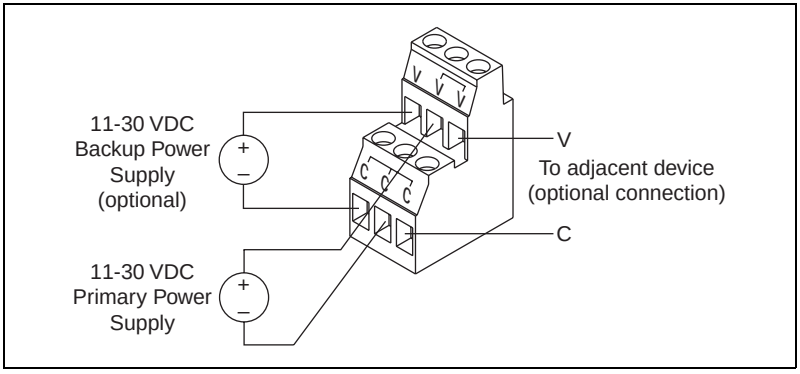


Figure 2-10. FP-3000 Power Connector Pinout

Connect the primary power supply to the center V and C pair with the positive and negative wires on your power cable in the V and C terminals, respectively. You can connect an optional backup power supply to the left V and C pair. The right V and C pair provides a convenient means of connecting power to the V and C terminals of a terminal base. Figure 2-10 shows this optional connection.

If your field I/O devices need to be powered separately, you can use the terminals provided on each terminal base for such power supply connections. Refer to the documentation that came with your terminal base and I/O module for more information on powering your field I/O devices.

Calculating Power for a FieldPoint Bank

The power requirements for a FieldPoint bank that uses an FP-3000 network module are calculated as follows:

$$Power = 6 \text{ watts} + 1.15 * \sum(\textit{I/O Module Consumption})$$

This is the amount of power the network module consumes from the power supply to power itself and the I/O modules. It does not include any power consumed by devices that you wire to the terminal bases.

The operating instructions for each FieldPoint I/O module contain power consumption information.

Power-On Self Test (POST)

The power-on self test (POST) is a test suite that the FP-3000 performs at power up to verify its own operational status. The test takes several seconds. The test is non-invasive and therefore does not affect the operation of the network, nor does it affect any of your field wiring connected to the terminal bases in the bank.

If the self-test suite fails, the FP-3000 does not participate in the network communication traffic, eliminating potential conflicts with the other banks in your network.

The FP-3000 indicates POST failure through the **POWER** and **STATUS** LEDs. Refer to the section *LED Indicators* for more information.

Connect the FP-3000 to the Fieldbus Network

The FP-3000 can be one of 32 devices connected to a Fieldbus network. The connection is made through the 9-pin male Dsub Fieldbus connector on the FP-3000, shown in Figure 2-11.

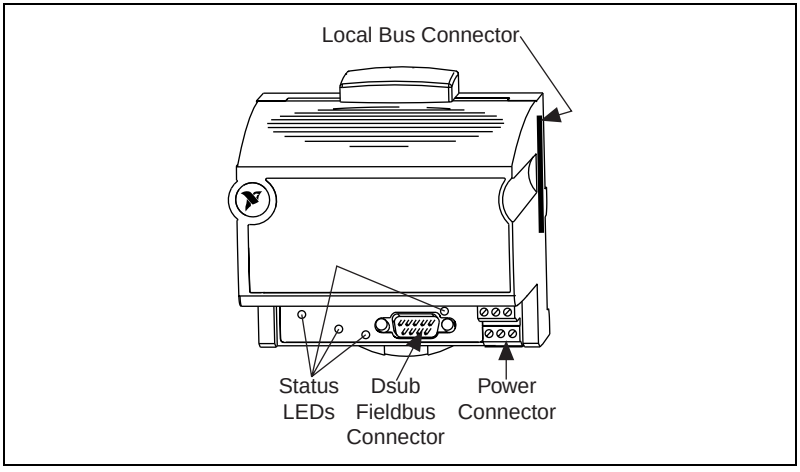


Figure 2-11. Fieldbus Connectors on the FP-3000

Use a Fieldbus cable with a 9-pin female Dsub connector to connect the FP-3000 to a properly terminated Fieldbus network. Refer to the Fieldbus Foundation *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium Application Guide* for specific information about wiring and installing a Fieldbus network. If you want to make your own Fieldbus cable, refer to the *Fieldbus Standard for Use in Industrial Control Systems, Part 2*, ISA-S50.02.1992. The FP-3000 Fieldbus connector pinout is shown in Figure 2-12.

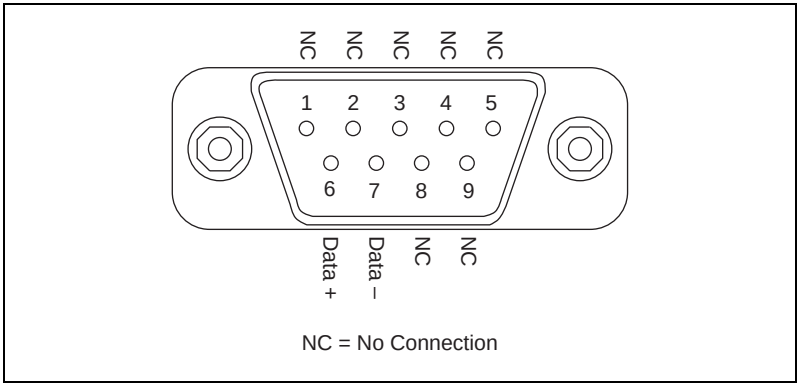


Figure 2-12. FP-3000 Connector Pinout

LED Indicators

The FP-3000 has four LED indicators: **POWER**, **NETWORK**, **PROCESS**, and **STATUS**. Figure 2-13 shows the LEDs on the FP-3000.

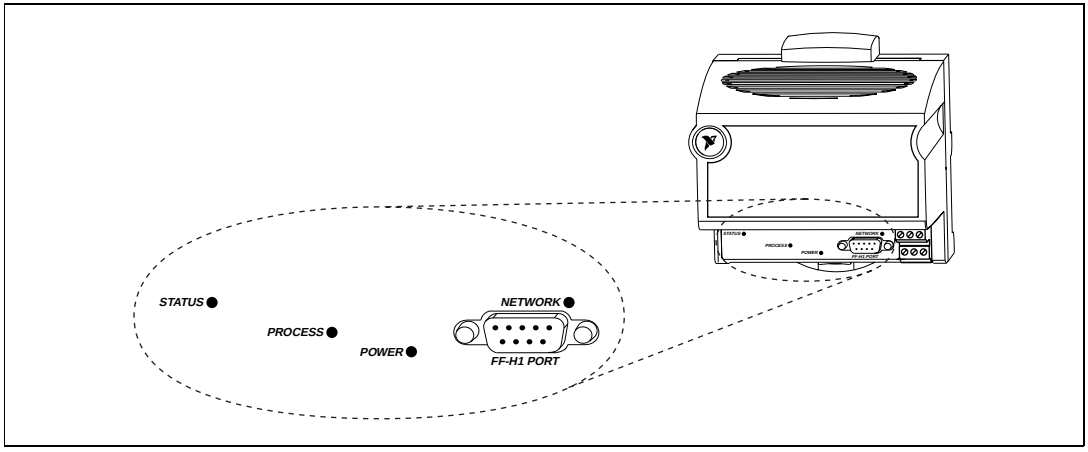


Figure 2-13. LEDs on the FP-3000

When power is applied, the **POWER** LED blinks green for approximately seven seconds during the power on self test. If the self test passes, the **POWER** LED turns solid green and the **READY** LEDs on each I/O module are lit green. If the self test fails, the **POWER** LED is lit red and the module enters an inactive state.

The multicolored **PROCESS** LED is used to indicate the current state of the processes being controlled by the FP-3000. When a PID function block on the FP-3000 module is in initialization, the light flashes green. When all the executing PID blocks on the FP-3000 are in Target mode, the light remains lit solid green. Any active alarm of priority greater than eight results in the light being lit red. For more information on PID blocks, refer to the section *PID (Proportional–Integral–Derivative)* in Chapter 4, *Block Reference*.

Table 2-1 describes the **NETWORK** LED states.

Table 2-1. Description of Fieldbus NETWORK LED States

NETWORK LED State	Meaning
Off	Fieldbus port not receiving data.
Flashing green	Fieldbus port is currently the Link Active Scheduler on the Fieldbus segment. The FP-3000 module can control communications on the Fieldbus.
Steady green	Fieldbus port is a basic device. The FP-3000 cannot control communication on the Fieldbus.
Flashing red and green	Fieldbus port is seeing traffic but is at a default Fieldbus network address. You need to assign a permanent network address through a Fieldbus configurator.
Steady red	Fieldbus port encountered fatal network error. Check the Fieldbus network connections.

The red **STATUS** LED is lit when the FP-3000 detects a failure. If **STATUS** is not lit, the FP-3000 has not detected a failure. The FP-3000 indicates specific error conditions by flashing **STATUS** a specific number of times. Table 2-2 describes the **STATUS** LED flashing sequences and the corresponding error conditions.

Table 2-2. STATUS LED Flashes and Corresponding Error Conditions

Number of Flashes	Error Condition
0 (stays lit)	Configuration has changed and has not been stored in static memory.
1	Parameter storage of nonvolatile and static parameters has been lost. Re-enter all stored parameters into the module. You can do this by re-downloading a saved configuration over the Fieldbus.
2	The FP-3000 detected an error in the terminal bases in the bank or identified a module in an illegal state. Verify that the protective cover is on the local bus connector of the last terminal base and that none of the pins of that connector are touching or bent. Verify that there are no more than nine terminal bases in the bank and that no terminal bases were added to the bank while power was applied.

HotPnP (Hot Plug and Play)

The HotPnP feature simplifies system installation, configuration, and maintenance. With the HotPnP feature, you can remove or insert I/O modules into the FieldPoint terminal bases while power is on, even if the system is already engaged in network activity. You do not need to power down the entire system or even a bank to insert, remove, or replace I/O modules. In addition, you do not need to change the operation of the host computer or software to use the HotPnP feature.

Caution To avoid damaging the network module and the terminal bases, make sure that you do not add or remove terminal bases while power is applied to the bank. An I/O module can be hot-inserted only if an empty terminal base is already available in the bank.

HotPnP During Operation

You might need to insert or replace one or more I/O modules in a bank while your system is operational (power is on and the network might or might not be active).

While one or more new or replacement I/O modules in a bank are being serviced by the HotPnP feature, the other I/O modules in the bank remain fully operational and accessible on the network without any interruptions.

As soon as the FP-3000 configures the new I/O module through the HotPnP service, that I/O module becomes automatically accessible on the network.

Inserting New I/O Modules

When a new I/O module is inserted, the FP-3000 automatically configures the I/O module to factory default settings. This configuration is accomplished without any intervention from the host computer or software.

Replacing I/O Modules

The host computer can detect missing I/O modules through the block alarm on the associated function blocks.

When a new I/O module is connected in place of one that was removed, the FP-3000 first verifies that the replacement I/O module is compatible with the one that was removed. If the I/O module is either the same as or compatible with the one removed, the FP-3000 configures the replacement I/O module with its predecessor's configuration and output value settings.

Hot-swap a module only with a compatible module. If you hot-swap a module with an incompatible module, the associated function blocks must be entirely reconfigured.

Updating the FP-3000 Firmware

As the FP-3000 evolves, National Instruments will release updates to the module that contain new features. These new features will include support for new types of FieldPoint I/O modules as they are released, as well as new function blocks and other enhancements. To update the firmware on an FP-3000, the FP-3000 Update utility (provided with the FP-3000) must be on a machine running the NI-FBUS Communications manager. You need to use the National Instruments AT-FBUS or PCMCIA-FBUS card on your PC or laptop computer. The new firmware features will be described by a new version of the Device Description.

Caution Updating the firmware on the FP-3000 may cause all FP-3000 configuration settings to be lost, depending on the degree of change in the firmware. You should make sure that all settings for the FP-3000 have been saved in your PC configurator before you update the firmware so that you can restore the settings after you update the firmware.

It is possible for two FP-3000 modules with different versions of the firmware and different DDs to co-exist on the same Fieldbus segment or a

Fieldbus system. You do not need to update all the FP-3000 modules with the new firmware. Follow these steps to update the firmware:

1. Select **FP-3000 Update Utility** from the start menu of the host PC.

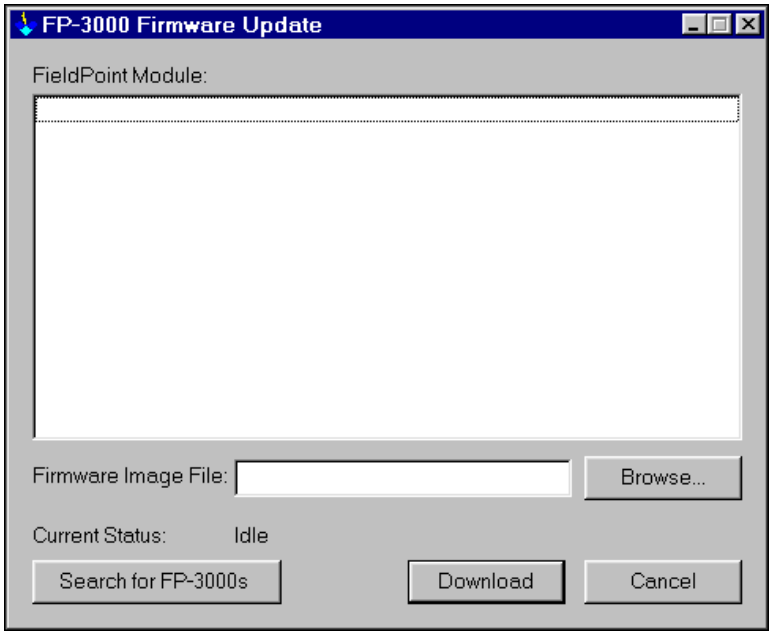


Figure 2-14. FP-3000 Firmware Update Dialog Box

2. Click on the **Search for FP-3000s** button. This causes the update utility to search every Fieldbus segment on the host PC for FP-3000 modules. Located modules are displayed as shown in Figure 2-15.

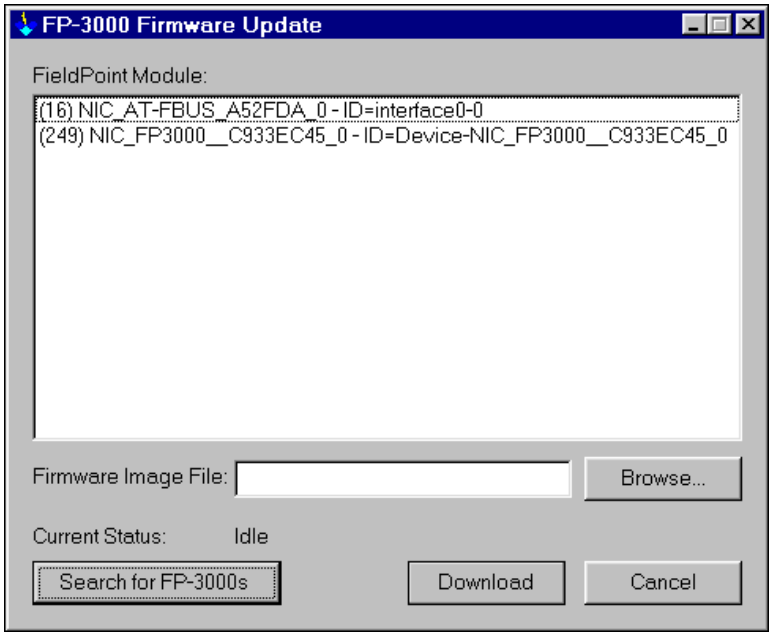


Figure 2-15. FP-3000 Search Dialog Box

3. Select the FP-3000 module that needs to be updated, and enter the path to the firmware image file.

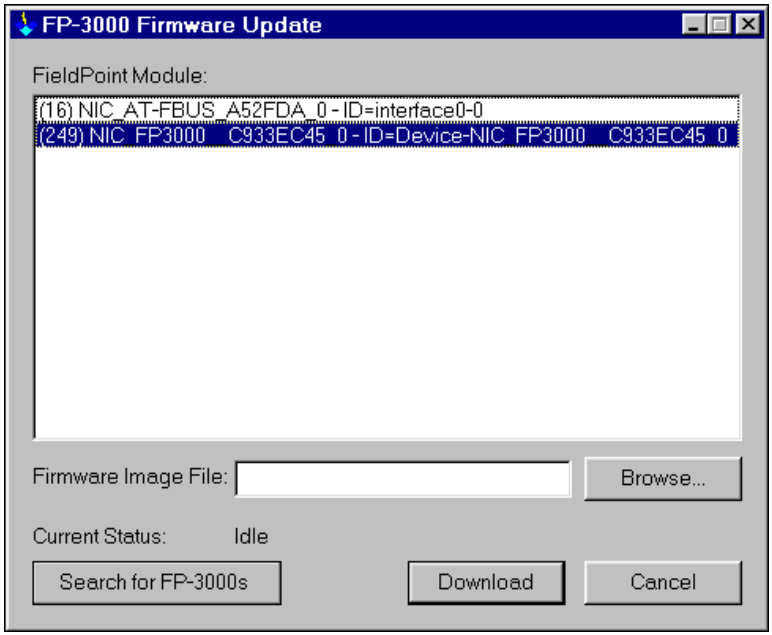


Figure 2-16. Select FP-3000 Module Dialog Box

4. Click on the **Download** button of the update utility. This process takes about 15 minutes. At the end of the process, the FP-3000 is updated to include the new features. At the end of the update process, the configuration information in the FP-3000 is cleared.

3

Example Applications

This chapter provides examples that show you how to configure the FP-3000 to perform common tasks, including reading from a 4–20 mA current loop device, taking temperature readings from a thermocouple module, and controlling the output current through an analog output module. This chapter also provides information about hardware and software configuration.

These examples assume you have the NI-FBUS Configurator; however, you can use any Fieldbus configuration utility capable of writing function block parameters, configuring linkages between parameters, and configuring function block schedules. If you are not using the NI-FBUS Configurator, refer to the documentation that came with your configurator for more details on how to perform software configuration-related tasks.

Before you run these examples, install the FP-3000 and the I/O modules. Connect the FP-3000 to the Fieldbus network and power it on. Start the NI-FBUS Configurator on your PC. Your configurator should show the FP-3000. For more information on installing the FP-3000, refer to Chapter 2, *Installation and Configuration*.

Initial Power On: Assigning Address and Device Tag

If you are powering on the FP-3000 for the first time, you need to perform some extra steps before you try these examples. You must assign each FP-3000 a unique device tag and network address before it can become operational. If you are using the NI-FBUS Configurator, the configurator automatically assigns a network address to the FP-3000 when it powers up. It also assigns a tag, which you can change if necessary by right-clicking on the device and choosing **Set Tag**. The process of automatic address assignment and tagging can take a few minutes. After the FP-3000 has a network address and tag, you can perform any of these examples. If you are not using the NI-FBUS Configurator, refer to the documentation that came with your configurator for information about setting the network address and device tag.

Example 1: Converting a 4–20 mA Pressure Sensor to Fieldbus Using FP-3000

One common application of the FP-3000 is interfacing to a conventional device, such as a 4-20 mA pressure sensor or a 4-20 mA temperature transmitter. This example helps you configure the FP-3000 to interface to a 4-20 mA pressure sensor. This example shows you how to instantiate an AI function block, assign a tag to the block, set up scaling parameters and range for the I/O channel, schedule the function block, and download the configuration to the FP-3000.

Getting Started

Example 1 requires the following materials:

- 4–20 mA sensor, such as a pressure sensor
- FP-AI-100, FP-AI-110, or FP-AI-111 module (this example assumes you are using the FP-AI-110 module)
- FP-3000 network module
- Host configuration system capable of instantiating function blocks on devices (such as National Instruments NI-FBUS Configurator)

Wire the 4–20 mA current loop into the current source input FP-AI-110 terminals. For channel zero, use terminals 1 and 17, as indicated on the wiring diagram on the front of the FP-AI-110 module. Make sure your current loop is powered and the sensor is operating normally.

Convert the Pressure Sensor Reading

Configure the software to perform the translation from the 4–20 mA sensor signal to the engineering units used on the Fieldbus.

Create Function Block

You must create a block for the FP-AI-110 since the pressure sensor is connected to a channel on the FP-AI-110 input module. To create a block for the FP-AI-110, follow these steps:

1. Right-click on the FP-3000 entry in your configurator.
2. Select **Instantiate New Block**. This causes a dialog box listing all the blocks supported by the FP-3000 to appear and allows them to be instantiated.

3. Select **FP AI 110 Block** from the list, then click on the **OK** button. This creates the correct analog input block on the FP-3000.

Assign a Tag to the New Block

By default, new blocks are created without a tag. To assign a tag, follow these steps:

1. Right-click on the new block, then select **Set Tag**.
2. Enter the tag you choose in the dialog box. The tag can be up to 32 characters in length and should not contain the dot (“.”) character.
3. Click on **OK**.

Select the Module and Channel

Determine the FieldPoint module number by counting each module in the order it is attached to the FP-3000, beginning with one. In this example, assume that the AI-110 is the only module connected to the FP-3000. Therefore, set the value of `FIELDPOINT_MODULE` to `1`. Since the transmitter is wired to the terminals associated with channel zero on the module, set the `FIELDPOINT_CHANNEL` parameter to `0`. To do this with an NI-FBUS Configurator, double-click on the block and select the **I/O Configuration** tab.

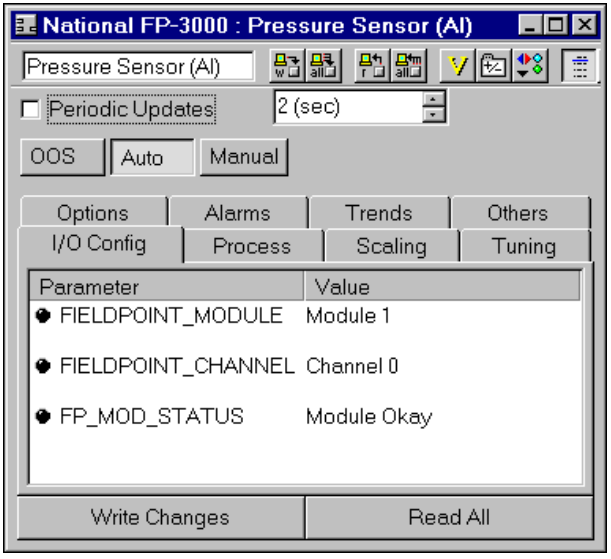


Figure 3-1. Setting the Channel Dialog Box

Set the Input Range

1. Find the FP_AI_110_RANGE parameter in the block.
2. Set the parameter to 3.5–21 mA, since this range most closely matches the 4–20 mA that you expect from your transmitter.

Scale the Reading

1. Set the XD_SCALE parameter, which tells the block the range of values to expect from the transducer. Go to the XD_SCALE parameter of the block and enter the following:

   ```
   XD_SCALE
       EU at 100%  0.020
       EU at 0%    0.004
       Units Index A
       Decimal     3
   ```

   This tells the AI block to expect readings in the range of 4 to 20 mA from the pressure sensor. The Decimal field is unused by the FP-3000, but may be used in some HMIs to determine the number of digits to display to the right of the decimal point.

2. Determine the pressure in your desired engineering units at 4 mA and at 20 mA. For example, suppose the sensor reads 10 inH20 (inches of water) at 4 mA and 250 inH2O at 20 mA. Go to the OUT_SCALE parameter of the block and enter the following:

   ```
   OUT_SCALE
       EU at 100%  250
       EU at 0%    10
       Units Index inH2O
       Decimal     3
   ```

3. Tell the block to use the scaling parameters. The block is flexible enough to either ignore scaling (Direct), use linear scaling (Indirect), or use square root scaling (Indirect square root). Since you want the block to use linear scaling, set the L_TYPE parameter of the block to Indirect.

The block converts the raw 4-20 value and reports it in engineering units through the PV and OUT parameters.

Set Up Scheduling

Before the block will operate, you need to schedule the block to execute. All Fieldbus function blocks (including function blocks on the FP-3000) execute according to a schedule. You can specify the order of function blocks in the schedule and the rate at which the schedule is repeated. To make the configurator create a schedule for your block, follow these steps:

1. Double-click on **Function Block Application** in the tree view of the configurator.
2. Drag the block from the tree view to the application view. The configurator automatically generates a schedule for the block that causes it to run every second (refer to the documentation that came with your configurator for information about changing the execution period).
3. To download this schedule to the device, select **Configure»Download Configuration**. The dialog box shown in Figure 3-2 appears. This dialog box enables the configuration to be downloaded. Go through the download process, as described in the documentation that came with your configurator. Since the block is not scheduled when you start the download, its actual mode does not change until the download is complete.

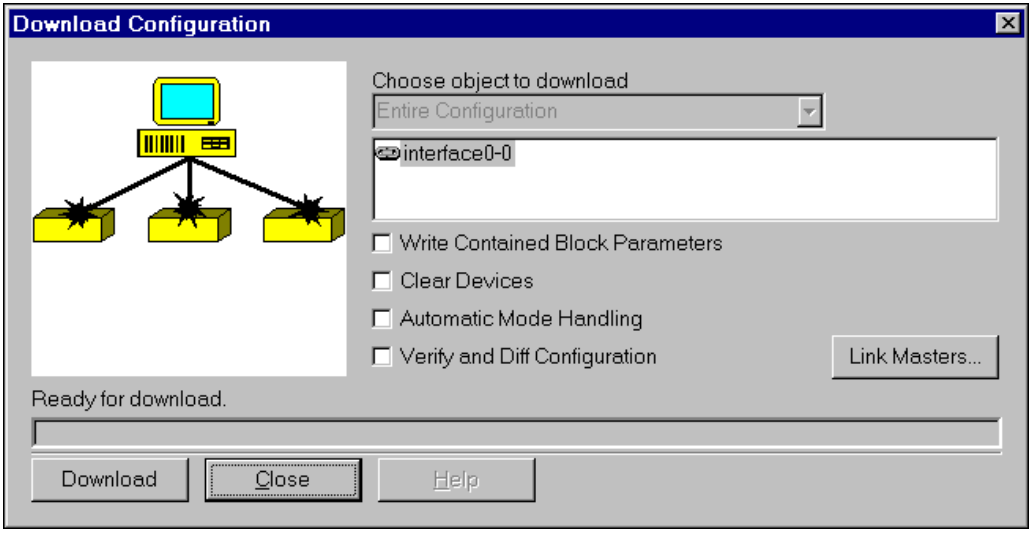


Figure 3-2. Downloading Configuration Dialog Box

Bring the Block Online

1. Go to the MODE_BLK parameter of the Resource block and set the TARGET to Auto.
2. Go to the MODE_BLK parameter of the block you created and set the TARGET to Auto.
3. Re-read the MODE_BLK parameter. The ACTUAL field should go to Auto. If it does not, refer to Appendix C, *Fieldbus Parameters*, for more information.

Once the block goes to Auto, it is fully operational. You can look at the value of OUT and see the pressure reading from the sensor in inches of water. The pressure reading can be displayed on an HMI, trended, or used for control (refer to the section *Example 2: Temperature Control with the FP-3000* for more information about using the reading for control).

Example 2: Temperature Control with the FP-3000

One application the FP-3000 is commonly used for is controlling temperature. A temperature control application might include a heating element and a temperature sensor and require temperature to be maintained at a constant level. Such an application would be well suited for PID control. In this example, the thermocouple measures the temperature in an enclosure, a PID block performs control, and the current output from an FP-AO-200 heats the heating element.

If you want to know how to get a thermocouple reading but are not interested in closed-loop PID control, perform only the steps in the section *Taking Temperature Readings*. After you complete those steps, the FP-3000 takes temperature readings. If you want to know how to output current to a simple device (like a resistive heating element) but are not interested in closed-loop PID control, proceed to the section *Controlling a Heating Element*, and perform the steps there. After you complete those steps, the FP-3000 controls output current.

Getting Started

Example 2 requires the following materials:

- Thermocouple or RTD input module (FP-TC-120 or FP-RTD-122)
- AO module such as the FP-AO-200 or FP-PWM-520 (this example uses the FP-TC-120)
- FP-3000 network module

Wire the thermocouple to channel zero of the FP-TC-120 module, paying attention to the polarity of the thermocouple wires. Channel zero is between terminals 1 and 2 of the FP-TC-120. Next, wire the heating element (say, a small light bulb or even a resistor) to channel zero of the FP-AO-200 module, which is between terminals 1 and 2 of the FP-AO-200.

You also need to connect a power supply to the V and C terminals of the FP-AO-200 module to source power on the output channel. Refer to the FP-AO-200 Installation Guide for more information.

Taking Temperature Readings

Create an FP-TC-120 Block

Once the hardware for control loop has been installed, you need to instantiate, or create, an I/O block for the thermocouple input channel. Since the thermocouple for this block is connected to a thermocouple input module, you must create a block. To instantiate an I/O block, follow these steps:

1. Right-click on **FP-3000** in the configurator.
2. Select **Instantiate New Block**. This causes a dialog box to appear that lists all of the blocks supported by the FP-3000 and allows them to be instantiated.
3. Select **FP TC 120 Block** from the list, then click on the **OK** button.

Assign a Tag to the New Block

By default, new blocks are created without a tag. To assign a tag, follow these steps:

1. Right-click on the new block, then select **Set Tag**.
2. Enter the tag you choose in the dialog box. The tag can be up to 32 characters in length and should not contain the dot (“.”) character.
3. Click on **OK**.

Select the Module and Channel

Determine the FieldPoint module number by counting each module in the order it is attached to the FP-3000, beginning with one. In this example, assume that the FP-TC-120 is the first module connected to the FP-3000. Therefore, set the value of `FIELDPOINT_MODULE` to `1`. Since the thermocouple is wired to the terminals associated with channel zero on the module, set the `FIELDPOINT_CHANNEL` parameter to `0`.

Set the Input Range and Thermocouple Type

1. Find the FP_TC_120_RANGE parameter in the block.
2. Set the parameter to 0–2048 K (degrees Kelvin).
3. Set the FP_THERMOCOUPLE_TYPE to the type of thermocouple you have connected (such as J or K type thermocouple).

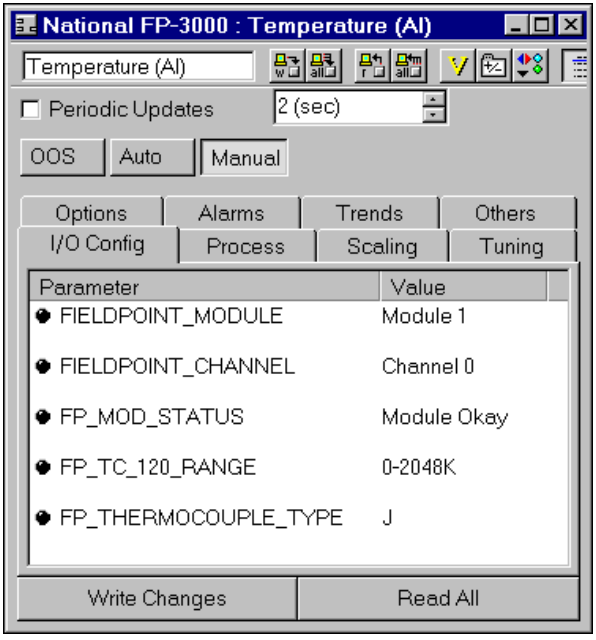


Figure 3-3. Set the Input Range and Thermocouple Type Dialog Box

Scale the Reading

1. Tell the block the range of values to expect from the transducer. Go to the XD_SCALE parameter in the block and enter the following:

```
XD_SCALE
    EU at 100%   2048
    EU at 0%     0
    Units Index  K
    Decimal      2
```

This tells the AI block to expect readings in the range of 0 to 2048 K from the thermocouple module.

You can avoid setting the XD_SCALE value manually if you set the CFG_OPTS option called **Automatically adjust XD_SCALE**. This allows the FP-3000 to copy the value from FP_TC_120_RANGE straight into the XD_SCALE parameter.

2. Determine the output scale. If you want to output the temperature in degrees Kelvin, you can set OUT_SCALE to the same values as XD_SCALE above. If you want to change to Celsius, you can do so by setting OUT_SCALE as follows:

```
OUT_SCALE
   EU at 100%  1775.00
   EU at 0%    -273.15
   Units Index °C
   Decimal     2
```

3. Tell the block to use the scaling parameters. The block is flexible enough to either ignore scaling (Direct), use linear scaling (Indirect), or use square root scaling (Indirect square root). Since you want the block to use linear scaling, set the L_TYPE parameter of the block to Indirect.

Set Up Scheduling

Before the block will operate, you need to schedule the block to execute. All Fieldbus function blocks (including function blocks on the FP-3000) execute according to a schedule. You can specify the order of function blocks in the schedule and the rate at which the schedule is repeated. To make the configurator create a schedule for your block, follow these steps:

1. Double-click on **Function Block Application** in the tree view of the configurator.
2. Drag the block from the tree view to the application view. The configurator automatically generates a schedule for the block that causes it to run every second (refer to the documentation that came with your configurator for information about changing the execution period).
3. To download this schedule to the device, select **Configure»Download Configuration**. A dialog box appears that enables the configuration to be downloaded. Go through the download process, as described in the documentation that came with your configurator. Since the block is not scheduled when you start the download, its actual mode does not change until the download is complete.

Bring the Block Online

1. Go to the MODE_BLK parameter of the Resource block and set the TARGET to Auto.
2. Go to the MODE_BLK parameter of the block you created and set the TARGET to Auto.
3. Re-read the MODE_BLK parameter. The ACTUAL field should go to Auto. If it does not, refer to Appendix C, *Fieldbus Parameters*, for more information.

Once the block goes to Auto, it is fully operational. You can look at the value of OUT and see the temperature reading from the thermocouple in degrees Celsius. The temperature reading is ready to be used for control.

If you are only interested in taking thermocouple readings and not interested in closed loop control, you are finished with this example.

Controlling a Heating Element

Instantiate an FP-AO-200 Block

Instantiate an I/O block for the FP-AO-200 channel to control the heating element. Since the heating element is connected to an FP-AO-200 module, you must create an FP-AO-200 block. To instantiate an I/O block, follow these steps:

1. Right-click on **FP-3000** in the configurator.
2. Select **Instantiate New Block**. This causes a dialog box to appear that lists all of the blocks supported by the FP-3000 and allows them to be instantiated.
3. Select **FP AO 200 Block** from the list, then click on the **OK** button.

Assign a Tag to the New Block

By default, new blocks are created without a tag. To assign a tag, follow these steps:

1. Right-click on the new block, then select **Set Tag**.
2. Enter the tag you choose in the dialog box. The tag can be up to 32 characters in length and should not contain the dot (“.”) character.
3. Click on **OK.**

Select the Module and Channel

Determine the FieldPoint module number by counting each module in the order it is attached to the FP-3000, beginning with one. In this example, assume that the FP-TC-120 is the first module connected to the FP-3000, and the FP-AO-200 is the second module. Therefore, set the value of FIELDPOINT_MODULE to 2. Since the heating element is wired to the terminals associated with channel zero on that module, set the FIELDPOINT_CHANNEL parameter to 0.

Set the Output Range

1. Find the FP_AO_200_RANGE parameter in the block.
2. Set the parameter to a current range of 0–0.021 A.

Scale the Output

1. Tell the block the range of values to output to the transducer module. Go to the XD_SCALE parameter in the block and enter the following:

   ```
   XD_SCALE
       EU at 100%  0.021
       EU at 0%    0
       Units Index A
       Decimal     2
   ```

   This tells the AO block to output readings in the range of 0 to 0.021 A to the FP-AO-200 module.

   You can avoid setting the XD_SCALE value manually if you set the CFG_OPTS option called **Automatically adjust XD_SCALE**. This allows the FP-3000 to copy the value from FP_AO_200_RANGE straight into the XD_SCALE parameter.

2. Determine the Process Variable (PV) scale. This scaling parameter is used to convert from the units of Set Point (SP) to percent of scale. For output function blocks like AO, SP is the value you want the block to output. For this example, set PV_SCALE to 0 to 100 percent. With these settings, a controller or operator changing the set point of this AO block writes the percentage of scale, with 100% being maximum output current.

   ```
   PV_SCALE
       EU at 100%  100
       EU at 0%    0
       Units Index %
       Decimal     2
   ```

In the case of Analog Output blocks, you do not need to tell the block to use the scaling parameters. The block will always use both XD_SCALE and PV_SCALE parameters.

Set Up Scheduling

Before the block will operate, you need to schedule the block to execute. All Fieldbus function blocks (including function blocks on the FP-3000) execute according to a schedule. You can specify the order of function blocks in the schedule and the rate at which the schedule is repeated. To make the configurator create a schedule for your block, complete the following steps:

1. Double-click on **Function Block Application** in the tree view of the configurator.
2. Drag the block from the tree view to the application view. The configurator automatically generates a schedule for the block that causes it to run every second (refer to the documentation that came with your configurator for information about changing the execution period).
3. To download this schedule to the device, select **Configure»Download Configuration**. A dialog box appears that enables the configuration to be downloaded. Go through the download process as described in the documentation that came with your configurator. Since the block is not scheduled when you start the download, its actual mode does not change until the download is complete.

Bring the Block Online

1. Go to the MODE_BLK parameter and set the TARGET to Auto.
2. Re-read the MODE_BLK parameter. The ACTUAL field should go to Auto. If it does not, refer to Appendix C, *Fieldbus Parameters*, for more information.

Once the block goes to Auto, it is fully operational. You can adjust the Set Point by writing a value between 0 and 100 to SP. The current flow through the heating element will vary.

If you are only interested in making FP-3000 output current and not interested in closed loop control, you are finished with this example.

PID Control

Instantiate a PID Block

Now that your input and output blocks are functioning, you can "close the loop" by creating a control block and putting the loop under automatic control. To instantiate a PID block, follow these steps:

1. Right-click on **FP-3000** in the configurator.
2. Select **Instantiate New Block**. This causes a dialog box to appear that lists all of the blocks supported by the FP-3000 and allows them to be instantiated.
3. Select **Proportional Integral Derivative Block** from the list, then click on the **OK** button.

Assign a Tag to the New Block

By default, new blocks are created without a tag. To assign a tag, follow these steps:

1. Right-click on **Proportional Integral Derivative Block**, then select **Set Tag**.
2. Enter the tag you choose in the dialog box. The tag can be up to 32 characters in length and should not contain the dot (".") character.
3. Click on **OK**.

Scale the PID

The PID has a PV_SCALE for scaling its input. The PV_SCALE should match the output scale (OUT_SCALE) of the AI block. To set the PID, enter the following:

```
PV_SCALE
   EU at 100%  1775
   EU at 0%    -273
   Units Index °C
   Decimal     2
```

The AO block takes a percentage range from the controller of 0 to 100 percent, so adjust the PID block to output that range. To set the OUT_SCALE parameter of the PID, enter the following:

```
OUT_SCALE
   EU at 100%  100
   EU at 0%    0
   Units Index %
   Decimal     2
```

Connect the PID to the AI and AO Blocks

1. Drag the new PID block to the Function Block Application window. All three of your blocks (AI, AO, and PID) should be in the window. If not, drag the remaining blocks into the window now.
2. Using the wiring tool, connect the `OUT` parameter from the AI to the `IN` parameter of the PID.
3. Connect the `OUT` parameter of the PID to the `CAS_IN` parameter of the AO.
4. Connect the `BKCAL_OUT` parameter of the AO to the `BKCAL_IN` parameter of the PID. Figure 3-4 shows what your connections should look like.

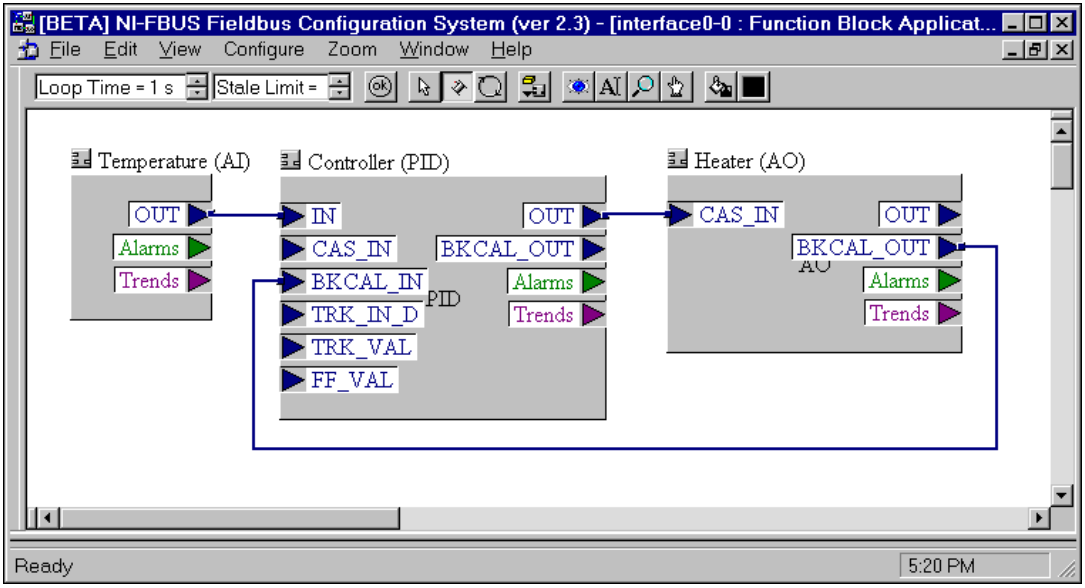


Figure 3-4. PID Block Connections Dialog Box

Download and Bring the Loop into Auto

1. Select **Configure»Download Configuration** to download your connection to the network. This establishes all the linkages that you "wired" in the previous step. It also schedules the PID function block, which has not yet been scheduled to execute.
2. When the download is complete, open the PID block.
3. Look at the `MODE_BLK` parameter. Set the `TARGET` to `Auto`.

4. Read the MODE_BLK parameter. The ACTUAL mode should read IMan, for Initialization Manual. This means the PID is not able to enter Auto because the AO block is not in Cascade mode.
5. Go to the AO block, and look at its MODE_BLK.
6. Set the TARGET to Cas and Auto (check both the Cascade and Auto boxes). This tells the AO to operate in Cascade if possible, and otherwise, to fall back to Auto.
7. Re-read the MODE_BLK parameter. ACTUAL should be Cas. If this does not happen, refer to Appendix C, *Fieldbus Parameters*, for more information.

Now, your loop should be running under automatic control. Verify this by reading the ACTUAL mode of the PID block. If it is Auto, the PID is trying to control the temperature. You can change the desired temperature by changing the Set Point (SP) parameter of the PID. Remember that the units of SP are the same as PV_SCALE for the PID, which in our example is degrees Celsius.

Tune the PID

Adjust the PID tuning constants to match the dynamics of your temperature process. A general description of how to tune a PID is beyond the scope of this document. However, the parameters to change in the PID block are GAIN, RESET, and RATE, and for temperature control, which is often fairly slow, the RATE parameter need not be used. You can adjust these constants, change the PID Set Point, and watch how the temperature changes over time.

Alarming

In the above example, it might be convenient to have FP-3000 generate an alarm whenever the temperature goes above 40° C. This behavior can be configured from the PID block or the AI block. This example uses the PID block.

To set a high limit alarm, follow these steps:

1. Open the PID block and locate the HI_PRI parameter. This is the priority of the high limit alarm.
2. Set the HI_PRI parameter to 2. Fieldbus alarms can range in priority from 0-15, with 0 being disabled and 1 meaning that the alarm is detected but not reported. All other priorities cause the alarm to be reported.

3. Set the HI_LIM parameter to 40. This is the high limit that triggers the high limit alarm. The units are defined to be the same as PV_SCALE, which is degrees Celsius.

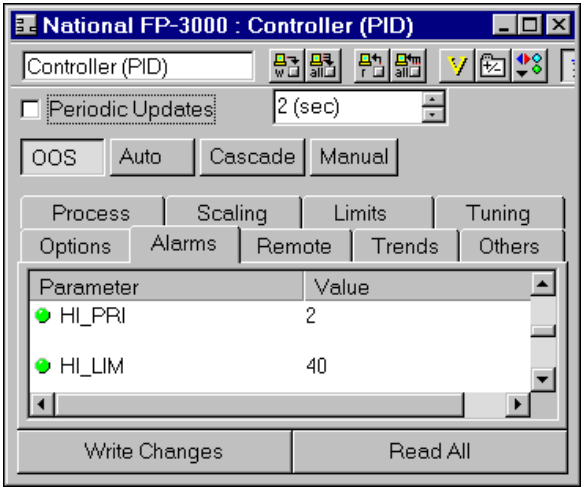


Figure 3-5. High Limit Alarm Parameters Dialog Box

4. Set up an interface card to receive the alarm. From the configurator, drag the icon that represents your interface card (it might be named something like “interface0-0”) onto the **Function Block Application** window. Connect the **Alarms** output of the PID to the **Alarms** input of the interface card.

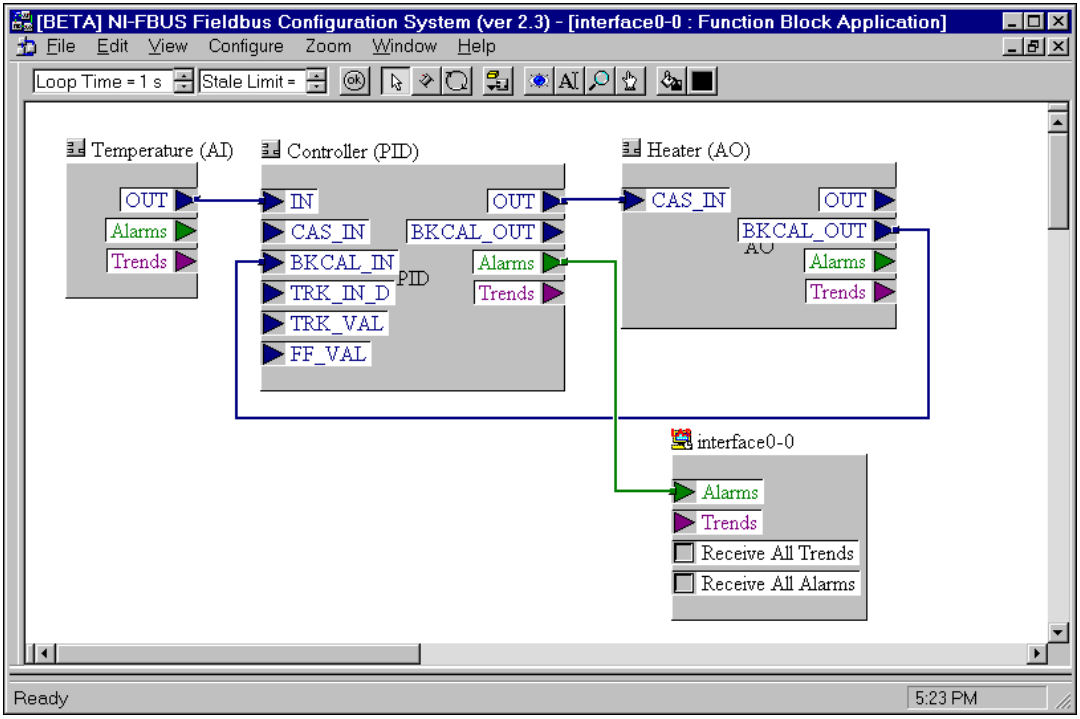


Figure 3-6. PID Alarm Connection Dialog Box

5. Download the configuration. The PID now detects a high limit alarm whenever the temperature exceeds 40° C, and the alarm is transmitted to the interface on your PC. You need a separate program (such as the Lookout HMI package from National Instruments) to receive, display, and acknowledge the alarms. You can verify that the alarms are being detected by the PID block by reading the `HI_ALM` parameter. The Alarm State changes, and the Alarm Timestamp is set when the alarm goes active.

4

Block Reference

This chapter describes the function blocks and the parameters supported by the FP-3000.

Block Overview

The FP-3000 consists of a number of blocks. A block is a predefined software module that runs on an FP-3000 and acts as a fundamental component of a control system. Each block has a number of parameters that can be used to adjust the configuration of that part of the system. These are referred to as contained parameters. In addition to contained parameters, some blocks have I/O parameters. I/O parameters of different blocks can be connected together to establish communications between blocks. Each block on an FP-3000 has an associated algorithm. There are three types of blocks: function blocks, transducer blocks, and the resource block.

Function Blocks

Function blocks implement the basic control algorithms that make up a control strategy. The Fieldbus Foundation has defined a set of ten fundamental (or elementary) function blocks and a set of nineteen advanced function blocks. The function blocks encapsulate a significant part of the control system behavior, thereby relieving a host of such tasks. The Fieldbus specification defines the parameters of each function block, how to make each parameter accessible to host system, parameters for configuring function blocks, and I/O parameters that can communicate to or from other function blocks in the system. For example, an Analog Input (AI) function block has parameters to scale a transducer value to engineering units. It also has alarm limits that can be configured by a host or a configurator. The AI detects and reports process alarms such as `HI_HI`, `HI`, `LO`, and `LO_LO`. The Fieldbus specification does not specify the algorithm for function blocks. For example, the specification does not define the actual equations to use in a PID function block. However, it does define all the parameters needed for configuration and operation of the PID, such as `RATE`, `GAIN`, `RESET`, and `MODE`. The execution of function blocks and the communication between function blocks on different devices are scheduled deterministically.

A PID control loop consists of one of each of the following function blocks: an Analog Input (AI) block to read the process variable in a device (such as a transmitter), a Proportional–Integral–Derivative (PID) block to compare the process value to the setpoint and make control decisions, and an Analog Output (AO) block to move an actuator in a device (such as a valve). The PID can be located in the transmitter, valve, or any other device (such as a controller). The execution of the AI, PID, and AO blocks is precisely scheduled on a time line. The communication of the process value from the AI to the PID and the communications between the PID and AO blocks are also scheduled and synchronized with the block executions.

The FP-3000 function blocks conform to the standard function blocks defined by the Fieldbus Foundation. In addition, the FP-3000 contains certain enhancements to the standard function blocks, such as AI, AO, DI, and DO, to permit easy configuration and diagnostics. The FP-3000 also has National Instruments defined function blocks. All the vendor-specific blocks and enhancements are defined using Device Descriptions to interoperate with other hosts and devices.

Resource Block

The resource block, defined by the Fieldbus specification, contains general information about the device. It also contains parameters to control the device as a whole, such as restarting the device or taking the device off-line. The resource block in the FP-3000 contains some enhancements to the standard resource block definition. For example, it includes a software version parameter that lists the version numbers of different modules and Fieldbus specifications supported.

Types of Function Blocks

The following types of blocks are supported by the FP-3000.

AI (Analog Input)

The AI block reads data from a single analog input channel on an FP-AI-110 module. This block performs simple filtering scaling of the raw data to engineering units from the input channel and supports limit alarming.

AO (Analog Output)

The AO block writes data to an analog output channel on an FP-AO-200 module. This block supports cascade initialization to allow upstream

control blocks to switch smoothly from manual to automatic mode. It also has a faultstate behavior that allows the block to react if communications fail between itself and the upstream block.

PID (Proportional–Integral–Derivative)

The PID block implements a PID control algorithm. When at least one PID block is present in the device, the Process LED reflects the state of the PID(s) present. In Fieldbus, a PID block must be connected to an upstream block (such as an AI block) and a downstream block (such as an AO block) before it can be used for control. These software connections are established by using a host Fieldbus configuration system, such as the NI-FBUS Configurator.

DI (Discrete Input)

The DI block reads data from discrete input channels. This block performs simple filtering and processing of the raw data from the input channel and supports limit alarming.

DO (Discrete Output)

The DO block writes to a discrete output channel. This block supports cascade initialization to allow upstream control blocks to determine the current state of the process before assuming control. It also has a faultstate behavior that allows the block to react if communications fail between itself and the upstream block.

CDO (Complex Discrete Output)

The CDO block serves the same purpose as the DO block and adds a number of parameters to support interlocking at three levels of priority.

Table 4-1. CDO Block Interlock Priorities

Input (Descending Priority)	Notes
Safeguard Close (`SAFEGUARD_CL`)	Safeguard Close takes priority over any other interlock input.
Safeguard Open (`SAFEGUARD_OP`)	Safeguard Open takes priority over every other interlock input and is used to force the block to an open state (`Discrete_State_1`).

Table 4-1. CDO Block Interlock Priorities (Continued)

Input (Descending Priority)	Notes
Binary Open/Close (BINARY_OP/BINARY_CL)	BINARY_OP only functions when ENABLE_OP has a value of Discrete_State_1. BINARY_CL only functions when ENABLE_CL has a value of Discrete_State_1. If both BINARY_OP and BINARY_CL are set and enabled, neither one takes effect.
Operator Command (OP_CMD_CXO)	OP_CMD_CXO is a contained bit string parameter that has a bit for Open and a bit for Close. Open only functions when ENABLE_OP has a value of Discrete_State_1. Close only functions when ENABLE_CL has a value of Discrete_State_1. If both Open and Close are set and enabled, neither one takes effect.

LOG (FieldPoint Log Block) (FP-3000 Specific)

The LOG block contains a log of error conditions and events detected by the device as it operates.

STAT (FieldPoint Statistics Block) (FP-3000 Specific)

The STAT block contains a set of parameters that can be used to examine how the device is performing. It contains statistics describing the performance of local function blocks and the network interface.

Function Blocks and FieldPoint Modules

The FP-3000 supports a wide variety of I/O channels, each with different types of configuration information. For example, a thermocouple channel includes parameters for thermocouple type and has ranges for different temperatures; a current loop channel includes parameters for filter frequency and has a different group of available ranges to chose from. Because of these differences in parameters, the FP-3000 has a block specific to each type of channel it supports. To use a thermocouple connected to an I/O channel on an FP-TC-120 module, the FP-3000 provides a FP-TC-120 AI Block. This block is a standard Analog Input block augmented with parameters specific to the thermocouple channel on the FP-TC-120.

Table 4-2. Function Blocks and FieldPoint Modules

Channel Type	Supported Module	Block Type
Analog Input	FP-AI-100	Fp Ai 100
	FP-AI-110	Fp Ai 110
	FP-AI-111	Ap Ai 111
	FP-TC-120	Fp Tc 120, Fp Tc 120 C
	FP-RTD-122	Fp Rtd 122
Analog Output	FP-AO-200	Fp Ao 200
	FP-PWM-520	Fp Pwm 520
Discrete Input	FP-DI-300	Fp Di 300
	FP-DI-301	Fp Di 301
	FP-DI-330	Fp Di 330
Discrete Output	FP-DO-400	Fp Do 400, Fp Cdo 400
	FP-DO-401	Fp Do 401, Fp Cdo 401
	FP-RLY 420	Fp Rly 420, Fp Crly 420

PID Control

In a Fieldbus network, a PID control loop is composed of three function blocks: an Analog Input (AI) block, a Proportional Integral Derivative (PID) block, and an Analog Output (AO) block. Figure 4-1 shows all three blocks and the interconnections between the blocks. The PID block's `IN` parameter is connected to the AI block's `OUT` parameter. The PID uses this linkage to determine the current value of the process variable it is controlling. The PID uses linkage from the PID's `OUT` parameter to the AO block `CAS_IN` parameter to adjust the AO block setpoint. To allow the cascade to be correctly initialized, a third back calculation linkage is created that allows the AO block to send its current setpoint back up to the PID block. These linkages must be established by using a PC-based Fieldbus configuration system, such as the NI-FBUS Configurator. For more information, refer to *Example 2: Temperature Control with the FP-3000* in Chapter 3, *Example Applications*.

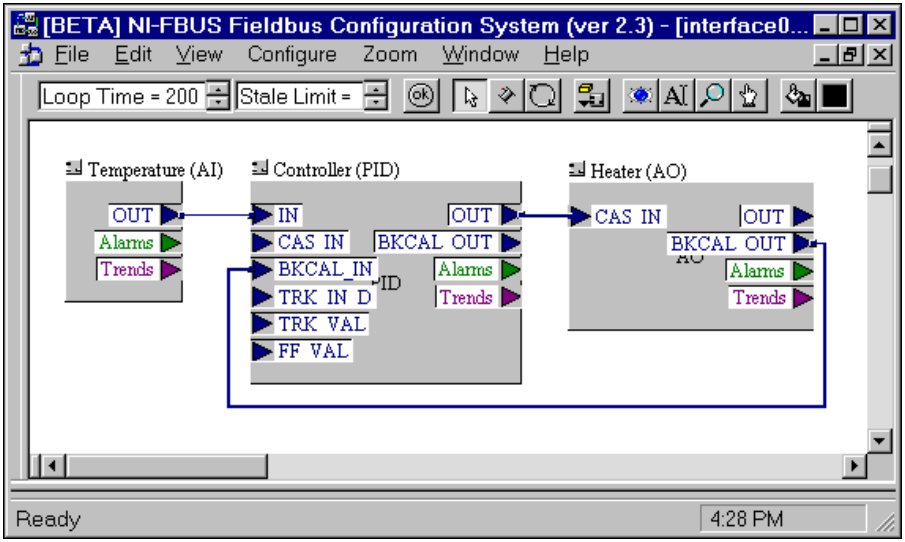


Figure 4-1. PID Function Block Application Dialog Box

Alarming

The Fieldbus network supports event notification messages from field devices like the FP-3000. Fieldbus function blocks use event notification messages to implement alarms and events. Alarms are used to report conditions that can either be active or inactive, such as the measured value of an AI block exceeding an alarm limit. The function block sends an event notification alarm to the host each time conditions transition between active and inactive. Events are notifications of one-time events as they are detected by the field device. An example of an event is the update event that is reported as a host application or operator modifies configuration parameters of the device.

Alarm Parameters

Each block contains a fixed set of alarms it can report, such as High Alarm or Deviation Alarm. The alarm parameter is a record describing the current state of the alarm or event. It contains a number of parameters the device uses to reveal the current state of the alarm. Following is a list of the meaning of each field in an alarm record.

UNACKNOWLEDGED

The UNACKNOWLEDGED subfield indicates the acknowledgment state of the alarm or event. A host application typically acknowledges the unacknowledged alarm when an operator sees and acknowledges the alarm.

ALARM_STATE/UPDATE_STATE

The current state of the alarm or event can be determined through the ALARM_STATE (for alarms) or UPDATE_STATE (for events) subfield of the alarm or event parameter. This parameter shows the active/clear state of alarms and the reported/unreported state of both alarms and events.

The first piece of state information in the state field is the active/clear state of the alarm. An alarm is considered to be active when the alarm condition is detected to be true. In the case of a limit alarm, the alarm is active when the process variable, such as the temperature being measured, is beyond the limit. The alarm state clears when the process variable returns within the limit plus a hysterisis factor specified in the ALARM_HYS parameter.

All blocks have one alarm known as the Block alarm. The Block alarm is considered active when any block error conditions (in the BLOCK_ERR parameter) are true. The Block alarm clears out when the last block error condition goes false.

For all alarms, the alarm condition is checked during each block execution. Events, on the other hand, are not considered to be active or clear, but simply one-time notifications.

The second piece of state information in the state field is the reported status of the alarm or event. When an alarm or event condition is reported to the host computer, an event notification message is broadcast on the bus if the alarm has a priority greater than 1. For alarms without priority parameters (Block alarms and events), the priority defaults to 2 and is always reported. To confirm the receipt of the event notification, the host computer responds with a confirmation message (different from the acknowledgment message discussed above). Until the device receives the confirmation message, the alarm or event is considered to be unreported. In the case where a block has no alarm linkage, the device waits to report the alarm or event until the linkage is established. If an alarm is unable to be reported to a host, the Active/Clear state of the alarm stays constant until the alarm can be reported.

TIME_STAMP

The time the alarm was detected by the FP-3000. In FOUNDATION Fieldbus, all devices share a common sense of time. This shared sense of time is used to timestamp alarm conditions as they occur, rather than when they are reported.

SUBCODE

For Block alarm, the subcode of the last error condition detected. The Block alarm, shared by all FOUNDATION Fieldbus function blocks in the FP-3000, is unique in that there are multiple conditions that can cause it to go active and clear. Any error condition reported in the `BLOCK_ERR` parameter of the block causes the block alarm to go active. The alarm does not clear until the last error condition in `BLOCK_ERR` has been resolved. To indicate which error condition the `BLOCK_ALM` is reporting, the `SUBCODE` subfield of the parameter is set to indicate the block error condition causing the fault. If additional error conditions are detected, the `SUBCODE` and `TIME_STAMP` are updated to reflect the latest condition detected, but the alarm will not be reported a second time until every error condition has been resolved.

VALUE

For limit alarms, the value of the parameter causing the alarm condition (the Process Variable). For update events, the index of the modified static parameter.

Status and Mode Handling Overview

Status and mode handling are crucial aspects of developing a distributed control application. Mode refers to the mode of operation of a function block; allowable modes depend on the type of block, but generally include Out of Service mode, Manual mode, and Auto mode.

Status refers to the quality of a variable communicated between two blocks. When a block receives a variable with bad status, it can affect its current mode of operation.

Status Handling

Parameters that can be communicated between blocks are composed of a value and a status. The value is the data to be communicated, and the status describes the quality of the data. When two blocks are logically connected

over the Fieldbus using a host configuration tool, the block that is sending data to the other block is called the publisher. The block receiving the data is called the subscriber. When communication is established from a publisher to a subscriber, the subscriber takes on the value and the status of the published variable. If communication is not established between the publisher and the subscriber, the subscriber has a status that reflects the lack of communication. Statuses themselves are composed of three subfields: the quality, the substatus, and the limit.

Quality

Table 4-3 describes the `Quality` subfields.

Table 4-3. Quality Values

Value	Meaning
`Bad`	The value is bad, the sensor is defective, or communication has not been established. The value should not be trusted by the receiver.
`Uncertain`	The quality of the data is unknown. This can be caused by errors or a lack of calibration in the physical I/O transducer. Blocks can generally be configured to treat values of `Uncertain` quality as either `Bad` or `Good` with the `STATUS_OPTS` parameter.
`Good_NonCascade`	The value is good and from a block that does not support cascade initialization. This status is also used when an alarm is active.
`Good_Cascade`	The value is good and from a block that supports cascade initialization.

Substatus

The `Substatus` field is used to describe more specifically the cause of the given quality. For example, a status with the quality of `Bad` might have a substatus of `Device Failure`, indicating that the value should not be trusted because it is from a device that has failed. Another common substatus is `Non-specific`. The `Non-specific` substatus is used when no other substatus applies. There are too many substatuses to explicitly list.

Limit

Table 4-4 describes the Limit subfield values.

Table 4-4. Limit Values

Value	Meaning
None	The value is not limited.
Low	The value is at a lower limit. This can be caused by a transducer limitation or setpoint limits.
High	The value is at a high limit. This can be caused by a transducer limitation or setpoint limits.
Constant	The value is at a fixed value and cannot move. This can be caused by the block supplying the value being in manual mode.

MODE_BLK Parameter and Mode Handling

The block's mode behavior is controlled with the MODE_BLK parameter. The MODE_BLK parameter contains four fields that allow the current mode of the block to be read and the desired mode for the block to be written. The four fields are Target mode, Actual mode, Permitted mode, and Normal mode.

Target Mode

The desired mode of execution for the block. An operator or process engineer normally writes this to put the block in the desired mode of operation.

Table 4-5. Target Modes

Desired Mode	Bit Set in Target Mode
Out of Service	Out of Service, and optionally any other valid target mode
Manual	Manual
Automatic	Automatic
Cascade	Cascade and Automatic

Table 4-5. Target Modes (Continued)

Desired Mode	Bit Set in Target Mode
`Remote Cascade`	`Remote Cascade` and `Automatic`. For more information on Remote Cascade operation, refer to the section *Fault State and Mode Shedding* in Appendix D, *Advanced Function Block Behavior*.
`Remote Output`	`Remote Output` and `Automatic`. For more information on Remote Output operation, refer to the section *Fault State and Mode Shedding* in Appendix D, *Advanced Function Block Behavior*.

Actual Mode

An indicator of the current mode of execution of the device. This is a read-only parameter. Normally, the actual mode of the block is equal to the target mode of the block. However, configuration errors or other conditions can cause it to differ from the target mode.

Table 4-6. Actual Modes

Mode (in descending priority)	Meaning
Out of Service `O/S` (0x80)	The block is idle and does not execute. If the resource block is out of service, all other blocks in the device are also out of service.
Initialization Manual `IMan` (0x40)	The upstream block in a cascade loop is preparing to enter `Auto` mode. This mode cannot be set as a target mode. It is used internally by control blocks as they establish cascade loops.
Local Override `LO` (0x20)	The block's faultstate or interlock capability is causing the block to override its normal output value. This mode cannot be set as a target mode.
Manual `Man` (0x10)	The output of the block is set by an operator through a write to the output parameter. No block processing other than writing to the I/O channel is performed.

Table 4-6. Actual Modes (Continued)

Mode (in descending priority)	Meaning
Automatic `Auto` (0x08)	The block operates normally with a setpoint specified manually through a write to the setpoint parameter.
Cascade `Cas` (0x04)	The block operates normally with a setpoint specified automatically through a connection from an upstream block to the `CAS_IN` (cascade input) parameter. Before the block can enter this mode, the cascade is initialized automatically to avoid windup.
Remote Cascade `RCas` (0x02)	The block operates normally with a setpoint specified automatically through a write from a host computer to the `RCAS_IN` (remote cascade input) parameter. Before the block can enter this mode, the cascade is initialized automatically to avoid windup.
Remote Output `ROut` (0x01)	The output of the block is set manually through a write to the `ROUT_IN` parameter. No block processing other than writing to the I/O channel is performed. Before the block can enter this mode, the cascade is initialized automatically to avoid windup.

Permitted Mode

A list describing the modes into which the block may be target. For example, in order to set a `Target` mode of `Manual`, the `Manual` mode bit must be set in the `Permitted` mode field. The `Permitted` mode field also has an effect on the way output blocks shed modes. For a description of mode shedding, refer to Appendix D, *Advanced Function Block Behavior*.

Normal Mode

The block's normal mode of operation is stored in the `Normal` field. This field is not used internally by the device, but is a guide for an operator.

FP-3000 Specific Parameters

CFG_OPTS

An option in the I/O function blocks to automatically set the scaling or alter the block behavior. Generally, this option can make configuration easier.

Table 4-7. Configuration Options

Option	Description
Automatically Adjust `XD_SCALE`	Enabling this option allows the block to automatically determine its `XD_SCALE` parameter, based on the physical I/O channel range, which is set by parameters such as `FP_AI_100` range.
Ignore Unconnected Interlock Inputs	By default, all interlock inputs on the CDO block must be good, or the block enters fault state. Enabling this option allows inputs with a status of `Bad::Not_Connected` to be ignored.

DEV_OPTS

A list of device-wide options that can be turned on and off at will. The current firmware supports only one option.

Table 4-8. Device Options

Options	Description
Disable `CFG_OPTS` in all I/O blocks	This bit disables all block configuration options in `CFG_OPTS`. It is not recommended that this bit be set since the additional behavior can make configuration easier.

EXECUTION_STATISTICS

A repository containing performance statistics for a given block. Use `EXECUTION_STATISTICS` to asses the performance of a given configuration and allow appropriate changes to be made.

Table 4-9. Execution Statistics

Statistic	Description
EXEC_COUNT	The number of times the block executed since the statistics were last reset.
EXEC_MISS_COUNT	The number of times the block failed to execute as scheduled since the statistics were last reset.
STALE_COUNT	The number of times the block received stale data since the statistics were last reset.
EVENT_COUNT	The number of events logged since the statistics were last reset.
RESET_TIME_STAMP	The time the statistics were last reset.

FIELDPOINT_CHANNEL

The FieldPoint I/O channel the block has been assigned to. Writing to this parameter updates the CHANNEL parameter appropriately.

FP-3000 determines the CHANNEL parameter automatically based on the FIELDPOINT_MODULE and FIELDPOINT_CHANNEL parameters. You do not need to set the CHANNEL parameter.

FIELDPOINT_MODULE

The FieldPoint I/O module containing the channel the block has been assigned to. FieldPoint modules are numbered, starting with one, at the I/O module closest to the FP-3000.

FP_AI_100_RANGE

Allows the range of a channel on a FieldPoint FP-AI-100 to be adjusted.

FP_AI_110_RANGE

Allows the range of a channel on a FieldPoint FP-AI-110 to be adjusted.

FP_AI_111_RANGE

Allows the range of a channel on a FieldPoint FP-AI-111 to be adjusted.

FP_AO_200_RANGE

Allows the range of a channel on a FieldPoint FP-AO-200 to be adjusted.

FP_CJC_SOURCE

Allows the cold junction compensation to be adjusted on a FP-TC-120 module.

Note Cold junction compensation is global to the entire module and affects every channel on the module.

FP_MOD_STATUS

The status of the FieldPoint I/O module associated with the function block.

Table 4-10. Module Status

Status	Description
`No Base`	There is no terminal base in the specified module position.
`Base, But No Module`	There is a terminal base in the specified module position, but no module is installed in the base.
`Unconfigured Module`	There is a module in the specified position, but the FP-3000 is unable to configure it.
`Module in Configuration`	There is a module in the specified position, the FP-3000 has attempted to configure the module, and the module is in the process of configuration.
`Module Okay`	There is a module in the specified position, and it is configured and operating correctly.
`Incorrect Module for Block`	There is a module in the specified position, but the block is incompatible with the module.

FP_NOISE_REJECTION

Allows the noise rejection filter of an Analog Input module to be adjusted.

FP_PWM_520_PERIOD

The period of the pulse width modulated waveform, in milliseconds.

FP_RTD_122_RANGE

Allows the range of a channel on a FieldPoint FP-RTD-122 to be adjusted.

FP_RTD_TYPE

Allows adjustment of the RTD type of a channel on an FP-TC-122 module.

FP_TC_120_RANGE

Allows the range of a channel on a FieldPoint FP-TC-120 to be adjusted.

FP_TC_120_CJ_RANGE

Allows the range of the cold junction compensation channel on a FieldPoint FP-TC-120 to be adjusted.

FP_THERMOCOUPLE_TYPE

Allows adjustment of the thermocouple type of a channel on an FP-TC-120 module.

LAST_BLOCK_EVENT

The last logged event detected by the block. Table 4-11 lists descriptions of the block events. This parameter is useful for debugging configuration errors because it points to the parameter in error. For example, if an AI block is in OOS mode because `L_TYPE` is not set, the `MGS` would say "Linearization Type Uninitialized."

Table 4-11. Block Events

Event	Description
CLASS	The type of event detected. **Configuration Error**: An error has been detected in the configuration of the block.This is usually due to an uninitialized static parameter. The block updates its target mode to Out of Service and posts a block alarm. **Operational Warning**: The block detected a non-critical event. The block continues to execute normally. **Operational Error**: The block detected a critical event. The block continues to execute in a higher priority mode. **Internal Error**: The firmware detected an internal error.
MSG	A message containing specific details describing the event.
BLOCK_IDX	The index of the block causing the event.
PARAM_IDX	The index of the parameter causing the event.
TARGET_MODE	The target mode of the block when the event was detected.
ACTUAL_MODE	The actual mode of the block when the event was detected.
TIME_STAMP	Time when the error was detected.

VERSION_INFORMATION

The revision of the firmware currently in use by the FP-3000. This parameter, present in the resource block, also contains the version numbers of the FOUNDATION Fieldbus specification documents used in the design of the FP-3000.

A

Configuring the FP-3000

The FP-3000 has three configuration switches accessible from an opening in the top of the module. These switches are shown in Figure A-1.

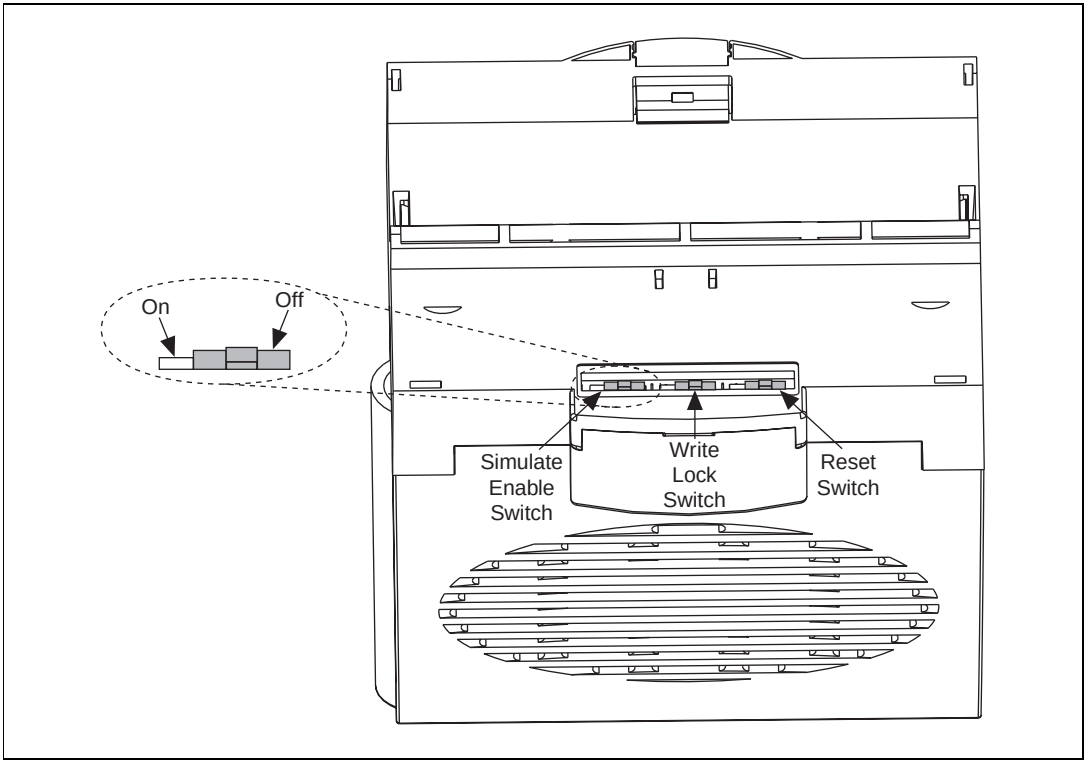


Figure A-1. Configuration Toggle Switches

Simulate Enable

When On, this switch allows simulation on I/O blocks to be enabled. The status of this jumper is shown in the Simulate Active bit in the resource block's BLOCK_ERR parameter. If the bit is set, the switch is On, and the device allows simulation to be enabled on I/O blocks.

Write Lock

When On, the device rejects writes to block configuration parameters. Linkages between blocks still function correctly.

Reset

When On, this switch causes the device to reset all configuration information to factory defaults on power up. To resume normal operation, this switch must be switched off and power to the device must be cycled a second time.

B

Troubleshooting

The FP-3000 is a powerful and highly flexible tool you can use to solve distributed I/O and control problems. Because of this flexibility, you might encounter problems getting the FP-3000 to perform the functions you want. This chapter helps you diagnose and solve common problems that you might encounter using the FP-3000. Problems you might encounter with the FP-3000 include Fieldbus communication problems, I/O module problems, and software configuration problems.

Fieldbus Communication Problems

The interface between the host computer and the FP-3000 is a FOUNDATION Fieldbus network. The network allows you to connect multiple devices (such as FP-3000s or other Fieldbus transmitters) and hosts together, with each device or host having a unique network address and a unique device tag. A master device on the Fieldbus constantly polls, or probes, empty network addresses to check for new devices. A number of problems can arise related to this networking scheme.

Setting Device Tag and Network Address

All Fieldbus devices such as the FP-3000 need a unique network address and a unique device tag before they can be fully operational. The FP-3000 ships without a device tag and with a default network address, which is a network address used temporarily to allow a device tag to be written by a host configurator. Once the device tag is written, a permanent network address can be assigned. If you are using the NI-FBUS Configurator, these steps happen automatically when the Configurator sees the FP-3000.

Table B-1 describes common Fieldbus communication problems and possible solutions.

Table B-1. Fieldbus Communication Problems

Problem	Solutions
FP-3000 is not visible on the Fieldbus from a host configurator	• Is the power LED on? If not, check your power supply and DC power wiring to the FP-3000. • Is the Fieldbus network light on? If not, your Fieldbus wiring is probably bad. • Check the switches on the back of the FP-3000. Make sure the **Reset** switch is not On. Refer to Appendix A, *Configuring the FP-3000*, to view a diagram of the configuration switches. • Is your Fieldbus host probing the network address of the FP-3000? If not, the FP-3000 will not be able to get on the network. Check your host documentation for how to make the host probe all network addresses. For NI-FBUS, use the **Interface Config** program, select **Advanced settings** for the interface port, and set **Num of unpolled nodes** to **0**, then restart NI-FBUS.
FP-3000 **NETWORK** LED is red (steady or flashing)	Check your Fieldbus wiring for short circuits or other electrical problems.
FP-3000 is visible on the Fieldbus from a host, but no blocks show up	If you are running NI-FBUS Configurator, and this is your first startup of FP-3000, wait a couple of minutes for the Configurator to complete setting the address and device tag of the FP-3000. When the process is complete, the blocks will show up. Otherwise, FP-3000 might be stuck at a default network address with no tag. Consult your host documentation on how to set the address and device tag.
All the blocks and configuration information are lost on a power cycle	Make sure the Reset switch at the back of the FP-3000 is set correctly. Refer to Appendix A, *Configuring the FP-3000*, for details.
FP-3000 does not execute the communication schedule when it becomes LAS	You must download a schedule to the FP-3000. If you are using NI-FBUS Configurator, make sure FP-3000 is in the list of devices to receive a schedule.

I/O Module Problems

FP-3000 offers diagnostic capabilities to help find problems with the I/O modules you have plugged in.

Table B-2. I/O Module Problems

Problem	Solutions
The green **READY** LED does not light when the module is plugged in.	Make sure the module is firmly seated. Also, make sure the terminal base is firmly attached to the terminal base on its left. Look for bent pins on the bottom of the module.
The red LED is lit on an I/O channel.	• TC module: The thermocouple is not connected or is broken. • AI module: The input wires are not connected or are broken. • AO module: The module cannot source as much current as the FP-3000 is requesting. This might be because the output wires are not connected or broken, or there could be some other electrical problem.

Software Configuration Problems

The FP-3000 software consists of a number of function blocks that you can instantiate, or create, from a host configuration utility. Each function block represents an input channel, an output channel, or some control behavior. In addition, the FP-3000 contains a Resource Block which allows you to configure some overall behavior for the FP-3000. All of these blocks contain parameters, which are values that you can set to configure things like channel ranges, scaling to engineering units, and failsafe behavior.

Some problems are generic to all blocks, and some problems occur only with certain types of blocks. Table B-3 lists problems that can occur in many types of blocks. Table B-4 lists problems that can occur only in the resource block.

Table B-3. Generic Software Configuration Problems

Problem	Solutions
Block will not leave OOS (Out of Service) mode, and BLOCK_ERR does not report any errors	• Make sure you have written a non-OOS mode to the TARGET mode of the block. • If this is a function block, make sure you have scheduled the block. Refer to Chapter 3, *Example Applications*, for information about how to download a schedule to the FP-3000. Function blocks must be scheduled to change modes. • The Resource Block might be OOS. This would force all function blocks into OOS mode. Set the MODE_BLK_TARGET parameter in the Resource Block to Auto, and make sure its ACTUAL mode changes to Auto. You do not need to schedule to Resource Block.
Block will not leave OOS (Out of Service) mode, and BLOCK_ERR parameter reads "Block Configuration Error"	• Make sure you have written a non-OOS mode to the TARGET mode of the block. • Look at the MSG field of the parameter LAST_BLOCK_EVENT on the block in question. This contains details on why the block cannot leave OOS mode. If this field is blank, the block might not be scheduled; function blocks must be scheduled to change modes. Refer to Chapter 3, *Example Applications*, for information about how to download a schedule to the FP-3000. If this field has a message, the message tells you which parameter of the block is misconfigured. Look up the correct use of the parameter in the Chapter 4, *Block Reference*, or Appendix C, *Fieldbus Parameters*, and reconfigure the parameter.

Table B-3. Generic Software Configuration Problems (Continued)

Problem	Solutions
Block leaves OOS mode but will not go into the exact TARGET mode	Your block is correctly configured, but some run-time condition is keeping the block from reaching its target mode. • If your block is connected to other function blocks as part of a Function Block Application, check the status of the input parameters. If any of these have a quality of Bad or Uncertain, examine the blocks the inputs came from to determine the problem. • If this block is not connected to other function blocks, but is operating standalone, check the BLOCK_ERR parameter to determine the problem. If BLOCK_ERR reports Input Failure or Output Failure, and this is an AI, AO, DI, DO, or CDO block, there is likely a problem with the I/O channel. For more information, refer to the section *I/O Module Problems*.
The actual mode of a block switches values	A block, such as a PID, may switch modes between IMAN and AUTO. This means that there is a communications problem between the PID and the AO blocks. • Check the schedule. The execution of the PID, AO, and the communication between them might be scheduled very close to each other. Space these events further apart and re-download the configuration. • Check if the stale limit is set correctly for the loop. If the block that is subscribing the data is executing at twice the rate of a block that is publishing data, the stale limit must at least be set to 2.
PID block output seems incorrect	Make sure that you have set the value of the tuning parameters like RATE, GAIN and RESET correctly. Recommended values for GAIN, RESET, and RATE are: GAIN / RESET / RATE Pressure control 1.2 3.33 0.8 Temperature control 3.0 25.0 10.0 Flow control 0.33 1.11 0.0 Level control 1.9 16.67 2.7

Table B-3. Generic Software Configuration Problems (Continued)

Problem	Solutions
Cannot access or view the parameters added by National Instruments	Make sure that the Device Descriptions are installed in the correct location. Refer to the section *Install the Device Description File* in Chapter 2, *Installation and Configuration*, if you are using NI-FBUS.
Cannot set the values of certain parameters	Most configuration parameters of function blocks can be changed only when the block is in `OOS` mode. Set the mode to `OOS` and then change the configuration parameters.

Table B-4. Resource Block Configuration Problems

Problem	Action
Cannot bring Resource Block into `Auto` mode	Make sure the Reset switch is disabled. You cannot bring the Resource Block into `Auto` with the Reset switch On (this prevents you from losing your entire configuration if you inadvertently leave the Reset switch On).
Cannot set the `WRITE_LOCK` parameter.	Make sure the Write Lock switch is On. Changing `WRITE_LOCK` is enabled only when the Write Lock switch is On. If you want to disallow configuration writes to the device, set the Write Lock switch to On, then write `Locked` to this parameter.

C

Fieldbus Parameters

ACK_OPTION

Allows alarms to be automatically acknowledged by the block with no outside intervention. This is useful if you are not interested in acknowledging certain alarms from a block.

ALARM_HYS

The amount a value must move off an alarm limit, in percent of scale, for the alarm to be considered clear. This helps prevent alarms from constantly "toggling" on and off when the process value is near the configured alarm limit.

ALARM_SUM

A summary of the status of alarms in the block. Allows alarms to be disabled.

ALERT_KEY

A value reported in alarm messages from the block that allows MMI applications to sort and filter alarms.

ALG_RUN_TIME

The length of time (in milliseconds) the block algorithm takes to run.

BAL_TIME

The time constant used by the integral term of the PID to obtain balance when the output is limited and the block is in `Auto`, `Cas`, or `RCas` mode.

BINARY_CL

An interlock input. When in `Discrete_State_1`, the output of the block is forced closed (`Discrete_State_0`). This interlock input has priority over all interlock inputs, except `SAFEGUARD_CL` and `SAFEGUARD_OP`. In the event `BINARY_OP` and `BINARY_CL` are in discrete state 1, both are considered to be in `Discrete_State_0`.

BINARY_OP

An interlock input. When in Discrete_State_1, the output of the block is forced open (Discrete_State_1). This interlock input has priority over all interlock inputs, except SAFEGUARD_CL and SAFEGUARD_OP. In the event BINARY_OP and BINARY_CL are in Discrete_State_1, both are considered to be in Discrete_State_0.

BKCAL_HYS

The amount a block's output value must move off a limit, in percent of scale, for the limit status to be turned off.

BKCAL_IN

The value from a downstream block's BKCAL_OUT parameter. This is used to initialize a control loop through cascade initialization. Cascade initialization allows smooth transfer for a control block from Manual to Auto mode. To bypass cascade initialization, this parameter can be left unwired and manually set to a status of Good, non-cascade.

BKCAL_OUT

A back-calculation value published to an upstream block in a control loop. The BKCAL_OUT parameter has the value of the block's current output. Before a cascade loop is initialized, the upstream block can use this value to smoothly transfer to loop control.

BLOCK_ALM

An alarm parameter used to report error conditions detected within the block, such as block Out of Service.

BLOCK_ERR

A list of error conditions detectable by the block. Table C-1 lists all active error codes.

Table C-1. Error Codes

Error Code	Code	Description
Other	0x0001	Undefined block error condition.
Block Configuration Error	0x0002	The block has detected an error in its configuration. This usually indicates a static parameter has been left uninitialized.
Link Configuration Error	0x0004	The logical connection between this block and another block is misconfigured.
Simulate Active	0x0008	For I/O function blocks, this indicates that simulation is enabled. For the resource block, this indicates that the simulate jumper has been set, allowing simulation to be enabled in other blocks.
Local Override	0x0010	The block has locally overridden the output value. This can be the result of an interlock or faultstate.
Device Faultstate Set	0x0020	The block's faultstate behavior is active.
Device Needs Maintenance Soon	0x0040	The device is reporting performance degradation that will soon require maintenance.
Input Failure/BAD PV Status	0x0080	Either the input transducer channel has reported a failure, or the input parameter from an upstream block has reported a failure. For an AI block, this could be caused by an open circuit being detected on the FP-AI-100 input module.
Output Failure	0x0100	The output transducer channel has reported a failure. For an AO block, this could indicate that the FP-AO-200 cannot drive the current request, perhaps due to an open circuit.
Memory Failure	0x0200	The storage for nonvolatile and static parameters was corrupted.
Lost Static Data	0x0400	The device was unable to restore the values of static parameters after a restart.
Lost NV Data	0x0800	The device was unable to restore the values of nonvolatile parameters after a restart.
Readback Check Failed	0x1000	The value read back from the output channel does not match the value the output channel was set to.

Table C-1. Error Codes (Continued)

Error Code	Code	Description
Device Needs Maintenance Now	0x2000	The device needs to be maintained now.
Power-Up	0x4000	The device has just powered up.
Out of Service	0x8000	The block is currently out of service.

BLOCK_RESET

Lets you reset the statistics or the configuration of an individual function block.

Table C-2. Block Reset Options

Option	Description
`Contained Parameters`	Sets all the contained parameters (standard and FP-3000 specific) of the function block to default values. This is useful to set a specific block to a known state without affecting the behavior of the other blocks in the FP-3000. The block should be in OOS mode to reset the contained parameters.
`Statistics`	Resets the various statistics counts associated with the block.

BYPASS

Allows the normal control algorithm to be bypassed if the `CONTROL_OPT` parameter's `Bypass Enable` option is selected. If control is bypassed, the PID uses its setpoint value as its output value and does not attempt to do any PID control.

CHANNEL

Used by I/O function blocks to select a physical I/O channel. This mapping is defined by the manufacturer. In the FP-3000, this parameter is automatically updated when the `FIELDPOINT_MODULE` and `FIELDPOINT_CHANNEL` parameters are modified.

CHECKBACK

A list describing the state of the interlock logic in the CDO block. Use `CHECKBACK` to determine how interlocks are operating.

Table C-3. Checkback States

State	Description
Safeguard Open	The SAFEGUARD_OP parameter is in Discrete_State_1, and the block has opened the output.
Safeguard Close	The SAFEGUARD_CL parameter is in Discrete_State_1, and the block has closed the output.
Binary Open	The BINARY_OP parameter is in Discrete_State_1, and the block has opened the output.
Binary Close	The BINARY_CL parameter is in Discrete_State_1, and the block has closed the output.
Safeguard Signal (LO)	The block has entered local override mode due to an active interlock.
Discrepancy in Open	Unused in FieldPoint.
Discrepancy in Close	Unused in FieldPoint.
Actuator Open	Unused in FieldPoint.
Actuator Close	Unused in FieldPoint.
Open Torque Exceeded	Unused in FieldPoint.
Close Torque Exceeded	Unused in FieldPoint.
Readback Simulated	Unused in FieldPoint.
Travel Time Exceeded	Unused in FieldPoint.
Local Lockout Active	Unused in FieldPoint.

CLR_FSTATE

Writing Clear to this parameter causes the device-wide faultstate to be cleared and output blocks to resume normal execution. Also see SET_FSTATE and FAULT_STATE parameters.

CONFIRM_TIME

The lower bound on the time the FP-3000 waits to send alert report messages if no confirmation is received from a host.

CONTROL_OPTS

A list of options used to adjust the way control blocks, such as the PID block, operate. The options are described in Table C-4.

Table C-4. Control Options

Options	Description
Bypass Enable	Allows you to set the BYPASS parameter and bypass the algorithm's control.
SP-PV Track in Man	Causes the setpoint to track the process variable in Man.
SP-PV Track in ROut	Causes the setpoint to track the process variable in ROut.
SP-PV Track in LO or IMan	Causes the setpoint to track the process variable in LO or IMan.
SP Track Retained Target	Permits the setpoint to track the input value of the retained target of the block. The retained target of the block is the lowest priority mode set in the target mode field of the MODE_BLK parameter of the block. For example, if the RCas bit is set in the target mode, the setpoint tracks RCAS_IN.
Direct Acting	Defines the relationship between changes to the PV and changes to the output. When set, an increase in the process variable causes the output value to be increased. When clear, a decrease in the process variable causes the output value to be decreased.
Track Enable	Enables tracking. If Track Enable is true, and TRK_IN_D is true, TRK_VAL overwrites the value at the output of the block.
Track in Manual	This enables tracking in Manual mode.
Use PV for BKCAL_OUT	When set, this uses the process variable as the value for BKCAL_OUT, instead of the setpoint.
Obey SP Limits if Cas or RCas	When set, this confines the setpoint to values within SP_HI_LIM and SP_LO_LIM, even when the setpoint comes from another function block.
No OUT Limits in Man	Unused in FieldPoint.

CYCLE_SEL/CYCLE_TYPE

Unused in FieldPoint.

DD_RESOURCE

Unused in FieldPoint.

DD_REV

The revision of the device description used by the device.

DEV_REV

The revision of the device.

DEV_TYPE

The manufacturer's model number for the device.

DV_HI_ALM

The current state of the deviation high alarm, along with a time and date stamp.

DV_HI_LIM

The deviation limit (between the PID block setpoint and process value) beyond which the deviation high alarm is considered active.

DV_HI_PRI

The priority of the deviation high alarm.

DV_LO_ALM

The current state of the deviation low alarm, along with a time and date stamp.

DV_LO_LIM

The deviation limit (between the PID block setpoint and process value) beyond which the deviation low alarm is considered active.

DV_LO_PRI

The priority of the deviation low alarm.

FAULT_STATE

The current status of the device faultstate. It can be set and cleared with SET_FSTATE and CLR_FSTATE. If it is set, all output blocks in the device initiate their own faultstate behavior.

FEATURE_SEL/FEATURES

The FEATURES parameter lists features supported by the device. Use the FEATURE_SEL parameter to manually enable and disable the supported features listed in the FEATURES parameter. The features are listed in Table C-5.

Table C-5. Feature Parameter Options

Option	Description
Unicode	The device supports strings in Unicode format. The FP-3000 does not support this feature.
Reports	The device supports event report messages for alarming. If this feature is not selected in the FEATURE_SEL parameter, the FP-3000 continues to detect alarms and events, but does not report them over the bus. In this case, the host must poll the alarm parameters to detect alarm conditions as they change.
Faultstate	The device supports Faultstate behavior for output blocks.
Soft Write Lock	The device supports locking of configuration of parameters with the WRITE_LOCK parameter in the resource block. With this feature selected and the WRITE_LOCK parameter written to "set," writes to all static configuration parameters are disallowed.
Hard Write Lock	The device supports locking of configuration parameters with a switch on the back of the FP-3000 (refer to Appendix A, *Configuring the FP-3000*). If Hard Write Lock is enabled, the switch disallows writes to all configuration parameters in the device, including FEATURE_SEL.
Out Readback	The device provides a way for the action of output transducers to be verified through a readback. The FP-3000 does not support this feature.
Direct Write	The device provides a manufacturer-specific way to directly write to I/O channels. The FP-3000 does not support this feature.

FF_GAIN

The gain by which the feed-forward input is multiplied before it is added to the output value of the control block.

FF_SCALE

The scaling parameter for the feed-forward parameter.

FF_VAL

The feed-forward value.

FIELD_VAL

The value from the input channel, in percent of scale.

FP_AUTOCONFIGURE

This parameter, present in the resource block, causes the FP-3000 to automatically configure itself. The FP-3000 detects all the I/O modules present and instantiates the appropriate I/O function blocks. It creates a function block for each I/O channel. It tags the function blocks and sets the contained parameters to appropriate defaults. The resource block must be set to OOS mode before you set the Autoconfigure option. If Autoconfigure is set on an existing configuration, the FP-3000 deletes all the existing blocks and linkages before creating new blocks.

FP_MOD_LIST

This parameter, present in the resource block, lists all the I/O module types that are currently plugged in.

FREE_SPACE

The percentage of free memory available on the device. This can be used when instantiating blocks to determine the remaining capacity of the FP-3000.

FREE_TIME

Unused in FieldPoint.

GAIN

The gain constant used by the PID in calculating the proportional component of the output.

GRANT_DENY

Allows MMI applications to determine access privileges for block parameters.

Note The device does not use this parameter to restrict parameter access itself. It is only for the benefit of host applications.

HARD_TYPES

A list of available channel types. As I/O modules are inserted and removed from the FP-3000 bank, bits in this field change to reflect the presence or absence of types of I/O channels.

Table C-6. Hard Types

Bitmask	Description
Analog Input	This bit is set if the FP-3000 has analog input channels available.
Analog Output	This bit is set if the FP-3000 has analog output channels available.
Discrete Input	This bit is set if the FP-3000 has discrete input channels available.
Discrete Output	This bit is set if the FP-3000 has discrete output channels available.

HI_ALM

The current state of the high alarm, along with a time and date stamp.

HI_HI_ALM

The current state of the high-high alarm, along with a time and date stamp.

HI_HI_LIM

The limit, in PV units, beyond which the high-high limit alarm is considered active.

HI_HI_PRI

The priority of the high-high limit alarm.

HI_LIM

The limit, in PV units, beyond which the high limit alarm is considered active.

HI_PRI

The priority of the high limit alarm.

IO_OPTS

A bitmask used to adjust the way I/O blocks operate. Table C-7 describes the operation bitmasks.

Table C-7. Operation Bitmasks

Bitmask	Description
Invert	In discrete blocks, this maps a physical state of `Discret_State_0` to `Discret_State_1` and maps every other physical transducer state to `Discret_State_0`.
SP-PV Track in `Man`	Forces the setpoint to track the process variable in `ROut`.
SP-PV Track in `LO` or `IMan`	Forces the setpoint to track the process variable in `LO` or `IMan`.
SP Track Retained Target	Forces the setpoint to track the input value of the retained target of the block. The retained target of the block is the lowest priority mode set in the target mode field of the `MODE_BLK` parameter of the block. For example, if the `RCas` bit is set in the target mode, the setpoint tracks `RCAS_IN`. For a list of modes in priority order, see Table 4-6 in Chapter 4, *Block Reference*.
Increase to Close	Remaps the block's scaling so that as the input increases, the output decreases.
Faultstate to Value	When set, the block's faultstate behavior sets the output value to the value in `FSTATE_VAL`. When clear, the block's faultstate behavior leaves the output value at its current setting.
Use Faultstate Value on Restart	Causes the output value of output blocks to go to faultstate value immediately after a device restart.
Target to `Man` if Faultstate Active	When set, this sets the target mode of the block to manual mode when faultstate goes active.
Use PV for `BKCAL_OUT`	When set, this uses the process variable as the value for `BKCAL_OUT`, instead of the setpoint.
Low Cutoff	When set, this enables the low cutoff parameter.

L_TYPE

The linearization type. This parameter affects the way the value from the transducer is linearized in the analog input block before it is presented as the block output. In all cases, the `FIELD_VAL` parameter behaves as follows:

$$\texttt{FIELD\_VAL} = \frac{(\texttt{transducer\_value} - \texttt{XD\_SCALE.EU0})}{\texttt{XD\_SCALE.EU100} - \texttt{XD\_SCALE.EU0}}$$

Table C-8. Linearization Types

Type	Description
`Direct`	The block output is directly taken from the transducer value: `OUT = transducer_value`
`Indirect`	The block output is scaled according to `OUT_SCALE` from the value in `FIELD_VAL`: `OUT = (FIELD_VAL(OUT_SCALE.EU100-OUT_SCALE.EU0))`
`Indirect Square Root`	The block output is scaled according to `OUT_SCALE` from the value in `FIELD_VAL`. Before the field value is rescaled, the square root is taken. $OUT = OUT\_SCALE.EU0 + \sqrt{\frac{FIELD\_VAL}{100}} \cdot (OUT\_SCALE.EU100 - OUT\_SCALE.EU0)$
`Uninitialized`	An invalid setting. The device reports a configuration error with an `Uninitalized L_TYPE`.

LIM_NOTIFY

A limit on the number of unconfirmed alarm/event notification messages the device can have active at once. This must be less than or equal to `MAX_NOTIFY`.

LO_ALM

The current state of the low alarm, along with a time and date stamp.

LO_LIM

The limit, in PV units, beyond which the low limit alarm is considered active.

LO_LO_ALM

The current state of the low-low alarm, along with a time and date stamp.

LO_LO_LIM

The limit, in PV units, beyond which the low-low limit alarm is considered active.

LO_LO_PRI

The priority of the low-low limit alarm.

LO_PRI

The priority of the low limit alarm.

LOW_CUT

With an L_TYPE of Indirect Square Root, this can be used to establish a floor for values from the transducer. Values below this floor are considered to be zero. The parameter can be disabled with the Low Cutoff option in the IO_OPTS parameter.

MANUFAC_ID

The ID of the manufacturer of the device. For National Instruments devices, it is 0x4E4943.

MAX_NOTIFY

The maximum number of unconfirmed alarm/event notification messages the device supports.

MEMORY_SIZE

Unused by FieldPoint.

MIN_CYCLE_T

The length of the shortest macrocycle the device supports.

MODE_BLK

Sets the operational and permitted modes of the block. Table C-9 describes the operational and permitted modes of the block.

Table C-9. Operational Modes

Mode	Description
Target	The desired mode of operation of the block.
	Out of Service (O/S). The block is out of service, block execution is suspended, and all output parameters take a status of Bad::OutOfService.
	Initialization Manual (IMan). The block is in the process of initializing a cascade. This is used for upstream (control) blocks when they are initializing for smooth transfer into Automatic mode.
	Local Override (LO). Faultstate or an interlock is active and causing the output value of the block to be overridden.
	Manual (Man). The output value of the block is set by the user.
	Auto (Auto). The output value of the block is set by the block to be equivalent to the setpoint (SP) parameter of the block.
	Cascade (Cas). The setpoint for the block is taken from the CAS_IN parameter. This mode cannot be entered before cascade initialization takes place.
	Remote Cascade (RCas). Like Cascade mode, in Remote Cascade mode the setpoint of the block comes from an outside data source. Unlike Cascade mode, in Remote Cascade mode the setpoint is sourced from the RCAS_IN parameter, which is written by a host application and not another function block.
	Remote Output (ROut). Remote Output mode is analogous to Remote Cascade mode, except that the remote host application directly sets the output of the block and not the setpoint. In the case of an analog output block, this bypasses setpoint rate and absolute limiting.
Actual	A bit reflecting the current state of operation of the block. This is a function of the target mode and the current conditions in which the block is executing.
Permitted	A bitmask indicating which modes are permitted target modes and which are not. This could be used by the plant operator to disallow certain modes the block would normally be permitted to have as a target mode.
Normal	Not used by the block, this can be used by an operator to store the normal mode of operation for the block in normal plant operations.

NV_CYCLE_T

The time interval in milliseconds in which nonvolatile parameters are committed to nonvolatile storage.

OP_CMD_CXO

The lowest level priority input. This can be used to allow the operator to activate interlock behavior with a write from the host application.

Table C-10. Command Parameters

Value	Description
`Close`	When this flag is set, the output of the block is forced to `Discret_State_0`. This interlock is overridden by every other interlock. If both `OP_CMD_CXO.Close` and `OP_CMD_CXO.Open` are set, they are both considered to be clear.
`Open`	When this flag is set, the output of the block is forced to `Discret_State_1`. This interlock is overridden by every other interlock. If both `OP_CMD_CXO.Close` and `OP_CMD_CXO.Open` are set, they are both considered to be clear.
`Stop`	Unused in FieldPoint.
`Enable 1, 2 ,3`	Unused in FieldPoint.

OUT

The current output value of the block.

OUT_HI_LIM

A limit for the maximum output value from a block in modes other than manual.

OUT_LO_LIM

A limit for the minimum output value from a block in modes other than manual.

OUT_SCALE

The scaling parameter used for the output parameter.

Table C-11. OUT_SCALE Parameter

Subfield	Meaning
EU_100	Engineering units value at 100 percent of scale.
EU_0	Engineering units value at zero percent of scale.
UNIT_INDEX	Actual engineering units code (such as mA).
DECIMAL	Number of digits a host shows to the right of the decimal for display purposes.

PV

The process variable being controlled by the process.

PV_FTIME

The filter time used in input blocks. For analog blocks, it is the time constant for a low pass exponential filter used to dump out rapid oscillations in the input value before using it as the process variable. For discrete blocks, it is the time the PV must remain constant after a change for the change to be reported.

PV_SCALE

The scaling parameter used by the process variable of the block. Converts from percent of scale to a process variable in engineering units. Contains the same subfields as OUT_SCALE.

RATE

The time constant for the derivative component of the PID block.

RCAS_IN

The cascade input set by a remote host. This is propagated to the setpoint of the block when it is in RCas mode. If the block is in RCas mode and this parameter is not updated in SHED_RCAS time (a parameter in the resource block), the block enters mode shedding. Mode shedding allows the block to degrade from RCas mode into some higher priority mode. Refer to Appendix D, *Advanced Function Block Behavior*, for more information about mode shedding.

RCAS_OUT

The back calculation output used by the host when establishing a Remote cascade loop.

RESET

The time constant for the integral component of the PID block. It is measured in seconds per repeat.

RESTART

Allows the user to restart the device remotely. Table C-12 lists restart values.

Table C-12. Restart Values

Value	Behavior
`Restart Resource`	Restarts the device.
`Restart to Defaults`	Restarts the device, restoring all parameter values to default values.
`Restart Processor`	Restarts the device as if the power was cycled.

Caution Using `Restart to Default` causes all your configured parameters in the FP-3000 to revert to their factory default settings.

ROUT_IN

The cascade input set by a remote host. This is propagated to the output of the block when it is in `ROut` mode. If the block is in `ROut` mode and this parameter is not updated in `SHED_ROUT` time (a parameter in the resource block), the block enters mode shedding. Mode shedding allows the block to degrade from `ROut` mode into some higher priority mode. For more information on mode shedding, refer to Appendix D, *Advanced Function Block Behavior*.

ROUT_OUT

This is the back calculation output used by the host when trying to establish a remote output loop. While the loop is being established, it is the current value of the output channel and can be used by the host to initialize for smooth transfer of control.

RS_STATE

The current state of the device. Table C-13 lists device states.

Table C-13. Device States

State	Meaning
`Start/Restart`	The device has just started a restart cycle.
`Initialization`	The device is performing startup diagnostics.
`Failure`	A hardware failure has been detected.
`On-Line Linking`	The device is online and waiting for new parameter linkages to be established.
`On-Line`	The device is online and in service.
`Standby`	The device is online, but currently out of service.

SAFEGUARD_CL

An interlock input. When in `Discret_State_1`, the output of the block is forced closed (`Discret_State_0`). This interlock input has priority over all other interlock inputs.

SAFEGUARD_OP

An interlock input. When in `Discret_State_1`, the output of the block is forced open (`Discret_State_1`). This interlock input has priority over all other interlock inputs except `SAFEGUARD_CL`.

SET_FSTATE

Allows the user to set the device faultstate to active. This, in turn, forces all output blocks into their own faultstate behavior.

SHED_OPT

Controls the way blocks enter mode shedding. Each option listed below has a companion `No Return` option. The `No Return` shedding options change the target mode of the device to the shed mode and prevent the device from re-entering `RCas` or `ROut` mode after the shed condition has ended.
Table C-14 lists shed conditions. Refer to Appendix D, *Advanced Function Block Behavior*, for more information about mode shedding.

Table C-14. Shed Conditions

Shed Mode	Behavior
Normal Shed	The block sheds into the next higher priority mode set in the permitted mode field of MODE_BLK.
Shed to Auto	The block sheds into automatic mode.
Shed to Manual	The block sheds into manual mode.
Shed to Retained	The block sheds to the next higher priority mode set in the target mode field of MODE_BLK.

SHED_RCAS

The shed time for the RCAS_IN parameter. If the block is in RCas mode and the RCAS_IN parameter has not been updated in SHED_RCAS time, the block performs mode shedding as determined by the SHED_OPT parameter.

SHED_ROUT

The shed time for the ROUT_IN parameter. If the block is in RCas mode and the ROUT_IN parameter has not been updated in SHED_RCAS time, the block performs mode shedding as determined by the SHED_OPT parameter.

SIMULATE

The simulate parameter is used to bypass the physical I/O channel and allow the block to operate normally, using a simulated I/O channel. For this feature to be enabled, you must set a switch on the back of the FP-3000. To see how to configure the switch, refer to Appendix A, *Configuring the FP-3000*.

SP_HI_LIM

The upper limit on the setpoint of the block. If the setpoint exceeds this value, the setpoint is considered to be SP_HI_LIM with a status that indicates that it is limited.

SP_LO_LIM

The lower limit on the setpoint of the block. If the setpoint is below this value, the setpoint is considered to be SP_LO_LIM with a status that indicates that it is limited.

SP_RATE_DN

The rate, in PV units per second, the setpoint can be moved downwards. If the setpoint moves faster than SP_RATE_DN, the block acts as if the setpoint is moving downwards at the maximum rate with a status bit that indicates that it is limited.

SP_RATE_UP

The rate, in PV units per second, the setpoint can be moved upwards. If the setpoint moves faster than SP_RATE_UP, the block acts as if the setpoint is moving upwards at the maximum rate with a status bit that indicates that it is limited.

ST_REV

ST_REV is incremented by one each time a static parameter is modified.

STATUS_OPTS

A collection of options that effects the status behavior of the block. Table C-15 lists the status options.

Table C-15. Status Options

Option	Meaning
IFS if Bad IN	Set the status of the block output to initiate faultstate if the IN parameter goes bad.
IFS if Bad CAS_IN	Set the status of the block output to initiate faultstate if the CAS_IN parameter goes bad.
Use Uncertain as Good	Treat the Uncertain status on an input parameter as if it was a Good status. Otherwise, Uncertain status is treated as bad.
Propagate Failure Forward	If the status of the IN parameter is Bad::Device_Failure or Bad::Sensor_Failure, propagate the failure to the OUT parameter and do not generate an alarm.
Propagate Failure Backward	If the status at BKCAL_IN or from the physical I/O channel is bad, propagate it to BKCAL_OUT and do not generate an alarm.
Target to Manual if Bad IN	Set the target mode of the block to Manual if the IN parameter has a bad status.
Uncertain if Limited	Produce an output status of Uncertain if the transducer value is limited.

Table C-15. Status Options (Continued)

Option	Meaning
Bad if Limited	Produce an output status of `Bad` if the transducer value is limited.
Uncertain if Manual Mode	Set the output status of the block to `Uncertain` if the block is in manual mode.
Do Not Select if Not `Auto` Mode	Set the output status of the block to `Do Not Select` if the block is not in an actual mode of `Auto`. This is useful for blocks upstream of the selector block.
Do Not Select if Not `Cas` Mode	Set the output status of the block to `Do Not Select` if the block is not in an actual mode of `Cas`. This is useful for blocks connected to a selector block.

STRATEGY

Used to identify groupings of blocks.

TAG_DESC

Used to describe the purpose of the block.

TEST_RW

Unused by the block algorithm. Used to test interoperability of reads and writes of different parameter types.

TRK_IN_D

Used to enable tracking of the output value to `TRK_VAL`. When this is true, the output value of the block takes on the value specified in `TRK_VAL`.

TRK_SCALE

The scaling parameter used for the value specified by `TRK_VAL`.

TRK_VAL

The track value of the block when tracking is enabled.

UPDATE_EVT

The current state of the update event, along with a time and date stamp. This event is issued whenever a static parameter is changed and ST_REV is incremented.

WRITE_ALM

The current state of the low alarm, along with a time and date stamp.

WRITE_LOCK

The software write lock for the device. When this is set to true, writes to all configuration parameters of all blocks are disallowed. The WRITE_ALM block alarm is active when writes are allowed and clear when they are disallowed.

WRITE_PRI

The priority of the write alarm.

XD_SCALE

The scaling parameter used to interpret values from the physical I/O channel. This is used to translate from a physical transducer value to a percent of scale.

Table C-16. Scaling Parameter Values

Subfield	Meaning
EU_100	Engineering units value at 100 percent of scale.
EU_0	Engineering units value at zero percent of scale.
UNIT_INDEX	Actual engineering units code (such as mA).
DECIMAL	Number of digits a host shows to the right of the decimal for display purposes.

D

Advanced Function Block Behavior

This appendix explains advanced features of function blocks that are unnecessary to establish simple control strategies. Use this information to diagnose problems in control strategies and to develop systems that implement supervisory control by a host computer.

Cascade Initialization

FOUNDATION Fieldbus provides a protocol called Cascade Initialization that allows a control function block to smoothly transition from `Man` to `Auto` mode. Cascade Initialization allows the PID algorithm to know the current setpoint of the AO block to balance the actual setpoint with the control's setpoint over time. Cascade Initialization is also used to prevent windup in the PID.

Parameter Connections for Cascade Initialization

Cascade initialization takes place between two blocks: an upstream controlling block, and a downstream controlled block. In a PID loop, the upstream block is the PID block, and the downstream block is the AO block. In the case of cascaded PID blocks, the upstream PID feeds a setpoint into a second PID that is acting as the downstream block. In both cases, the parameter connections are the same. The output (`OUT`) parameter of the upstream block is connected to the cascade input (`CAS_IN`) parameter of the downstream block. This connection controls the setpoint of the downstream block. To allow the upstream block to determine the current setpoint of the downstream block, you must also connect the backward calculation output (`BKCAL_OUT`) parameter of the downstream block with the backward calculation input (`BKCAL_IN`) of the upstream block. The connections are shown in Figure D-1.

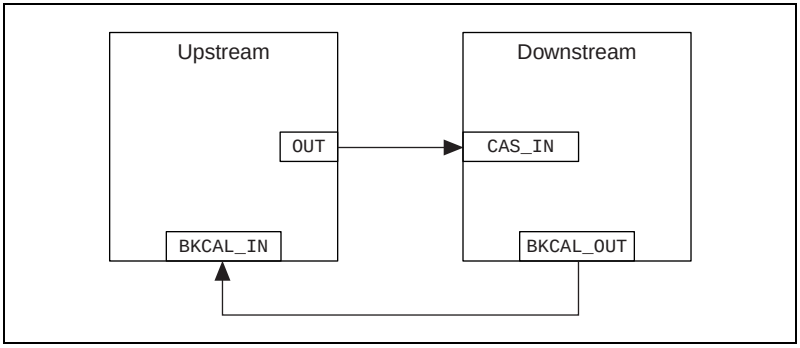


Figure D-1. Parameter Connections for Cascade Initialization

Mode and Status Behavior during Cascade Initialization

Cascade initialization is arbitrated through the status of the backward calculation path from the downstream block and the forward calculation path on the upstream block. If the upstream block publishes a status of `Good, Non-Cascade`, it does not support cascade initialization, and the lower block immediately transitions into a Cascade mode. This happens in the case where an Analog Input (AI) block is acting as the upstream block for an Analog Output (AO) block. Since the AI block does not have a back calculation input, it does not support cascade initialization.

If the upstream block does support cascade initialization, it publishes a status of `Good, Cascade` on its forward calculation output. This signals to the downstream block to begin the cascade initialization process as soon as it is able. If the downstream block is unable to begin cascaded control, it publishes a status of `Good Cascade, Not Invited` on its backward calculation output. This signals to the upstream block that the control path from the downstream block to the process has been broken. As soon as the ability to begin control is established, the downstream block publishes a status of `Good Cascade, Initialization Request` on its backward calculation output. This signals to the upstream block that it should initialize itself for cascade control. While the initialization request status is active, the downstream block is also publishing its current setpoint to the upstream block. This enables the upstream block to prepare for a smooth transfer to automatic control. While the upstream block is initializing itself for automatic control, it enters an actual mode of Initialization Manual (`IMan`). When it is ready to begin control, it publishes a status of `Good Cascade, Initialization Acknowledge` to signal that it is beginning cascade control. The lower block then enters Cascade mode.

To prevent windup, the control loop needs to be aware when it is unable to control the process. If the downstream block can no longer control the process, it reports a status of Bad to the upstream block. This breaks the cascade until automatic control can be resumed, in which case cascade initialization takes place again.

Remote Cascades

If a host application (rather than another block) provides the setpoint of a block, FOUNDATION Fieldbus provides the `Remote Cascade` mode. The remote cascade mode is equivalent to `Cascade` mode, except that the cascade input parameter is `RCAS_IN` instead of `CAS_IN`, and the back calculation output is `RCAS_OUT` instead of `BKCAL_OUT`. Unlike `CAS_IN` and `BKCAL_OUT`, which are input/output parameters, `RCAS_IN` and `RCAS_OUT` are contained parameters and can only be written by a host application. To allow the controlled block to enter `Remote Cascade` mode, the host application must act as the upstream block in the cascade initialization and implement the status handling described above.

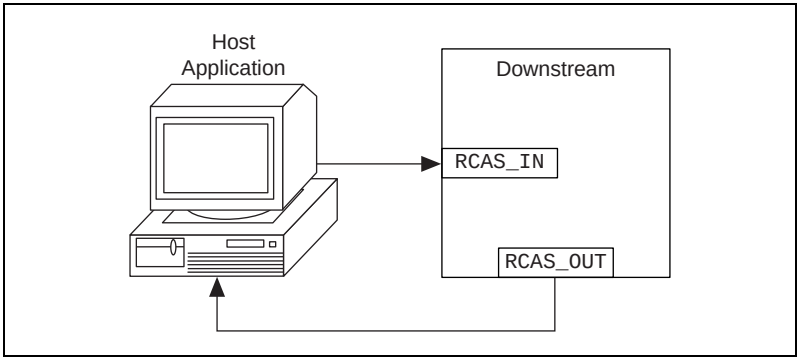


Figure D-2. Remote Cascade Model

There is a second remote mode in which a cascade must be initialized: Remote Output (`ROut` mode). Unlike `RCas` mode, where the block setpoint is set remotely, in `ROut` mode the block output is set by the host application. The back calculation output for `ROut` mode is `ROUT_OUT`, and the cascade input is `ROUT_IN`. Otherwise, cascade initialization proceeds normally.

Bypassing Cascade Initialization

If cascade initialization is unnecessary, you can bypass it by leaving the back calculation input of the upstream block unwired. By manually writing a status of `Good, Non-cascade` to the back calculation input, the block bypasses cascade initialization and immediately transitions to automatic control. The upstream block signals to the downstream block that cascade initialization has been bypassed by setting its output to a status of `Good, Non-cascade`.

Fault State and Mode Shedding

To allow for safe control of a process, even in the event the input sensors or control algorithms fail, FOUNDATION Fieldbus provides fault state and mode shedding. Fault state is used when an output block is in `Cascade` mode. Mode shedding is used in `Remote` or `Remote Cascade` modes.

Fault State

FOUNDATION Fieldbus output function blocks (AO, DO, and CDO) support a faultstate to deal with the case where the control of the output block has been lost while the block is in `Cascade` mode. If the block is in `Cascade` mode, and `CAS_IN` has a bad quality for longer than the time specified in the `FSTATE_TIME` parameter, the block enters faultstate. If the `Faultstate Use Value` option is set in the `IO_OPTS` parameter, the block uses the value in `FSTATE_VAL` as its output value. If the option is not set, it holds the value it had at the time the input went bad until the cascade can be reestablished.

Mode Shedding

For remote modes in which the cascade input is periodically written by a host application, FOUNDATION Fieldbus provides mode shedding to handle failure of the host application. For the `RCas` mode and `ROut` modes, there are two timeout parameters in the resource block: `SHED_RCAS` and `SHED_ROUT`. If the block is in a remote mode and the block input being used (`RCAS_IN` or `ROUT_IN`) is not updated within the timeout period, the block sheds to a higher priority mode. The action taken by the block when mode shedding occurs is defined by the `SHED_OPT` parameter, as described in Table D-1. The shed condition ends when the host writes the remote input parameter. If a normal return shed option is selected, the block attempts to enter the original remote mode. If a no return shed option is selected, the

target mode is changed at the time the block sheds mode, and the block does not attempt to enter the original lower priority mode when the shed condition ends.

Table D-1. Mode Shedding Options

Shed Option	Behavior on Timeout Detection
Shed to Manual	The block enters `Manual` mode.
Shed to Auto	The block enters `Auto` mode.
Shed to Retained	The block sheds to the next highest priority mode set in the target mode field.
Normal Shed	The block sheds to the next highest priority mode set in the permitted mode field.

E

Specifications

This appendix describes the specifications of the FP-3000 network module.

All FieldPoint network modules undergo extensive testing for operating under rugged environmental conditions that exist in industrial applications. FieldPoint network modules are designed and tested for immunity and susceptibility, as well as for emissions.

Network

FP-3000 ..... FOUNDATION Fieldbus H1

Integrity ..... Checksum

Power Supply Range ..... 11 to 30 VDC

Power Consumption ..... 6 watt + 1.15 * ∑(*I/O Module Consumption*)

Maximum Terminal Bases per Bank 9

Maximum Number of Banks per Fieldbus segment (without repeaters) 32

Environment

Operating Temperature ..... –40° to +60° C

Storage Temperature ..... –55° to +100° C

Relative Humidity ..... 5% to 90% noncondensing

Compliance

Electrical Safety ..... designed to meet IEC 1010

EMI Emissions/Immunity ..... CISPR 11

F

Technical Support Resources

National Instruments offers technical support through electronic, fax, and telephone systems. The electronic services include our Web site, an FTP site, and a fax-on-demand system. If you have a hardware or software problem, please first try the electronic support systems. If the information available on these systems does not answer your questions, contact one of our technical support centers, which are staffed by applications engineers, for support by telephone and fax. To comment on the documentation supplied with our products, send e-mail to `techpubs@natinst.com`.

Web Site

The InstrumentationWeb address is `http://www.natinst.com`.

From this Web site you can connect to our Web sites around the world (`http://www.natinst.com/niglobal/`) and access technical support (`http://www.natinst.com/support/`).

FTP Site

To access our FTP site, log in to our Internet host, `ftp.natinst.com`, as `anonymous` and use your e-mail address, such as `yourname@anywhere.com`, as your password. The support files and documents are located in the `\support` directories.

Fax-on-Demand Support

Fax-on-Demand is a 24-hour information retrieval system containing a library of documents in English on a wide range of technical information. You can access Fax-on-Demand from a touch-tone telephone at 512 418 1111.

E-Mail Support

You can submit technical support questions to the applications engineering team through e-mail at `support@natinst.com`. Remember to include your name, address, and phone number so we can contact you with solutions and suggestions.

Telephone and Fax Support

National Instruments has branch offices all over the world. Use the following list to find the technical support number for your country. If there is no National Instruments office in your country, contact the source from which you purchased your software to obtain support.

Country	Telephone	Fax
Australia	03 9879 5166	03 9879 6277
Austria	0662 45 79 90 0	0662 45 79 90 19
Belgium	02 757 00 20	02 757 03 11
Brazil	011 284 5011	011 288 8528
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Québec)	514 694 8521	514 694 4399
Denmark	45 76 26 00	45 76 26 02
Finland	09 725 725 11	09 725 725 55
France	0 1 48 14 24 24	0 1 48 14 24 14
Germany	089 741 31 30	089 714 60 35
Hong Kong	2645 3186	2686 8505
India	91805275406	91805275410
Israel	03 6120092	03 6120095
Italy	02 413091	02 4139215
Japan	03 5472 2970	03 5472 2977
Korea	02 596 7456	02 596 7455
Mexico (D.F.)	5 280 7625	5 520 3282
Mexico (Monterrey)	8 357 7695	8 365 8543
Netherlands	0348 433466	0348 430673
Norway	32 84 84 00	32 84 86 00
Singapore	2265886	2265887
Spain (Madrid)	91 640 0085	91 640 0533
Spain (Barcelona)	93 582 0251	93 582 4370
Sweden	08 587 895 00	08 730 43 70
Switzerland	056 200 51 51	056 200 51 55
Taiwan	02 2377 1200	02 2737 4644
United Kingdom	01635 523545	01635 523154
United States	512 795 8248	512 794 5678

Glossary

Prefix	Meanings	Value
m-	milli-	10^{-3}
M-	mega-	10^{6}

%	percent
°	degrees
0x	precedes a hexadecimal number
B	bytes
bank	The combination of one FieldPoint network module and one or more terminal bases and I/O modules.
C	Celsius
CISPR	International Special Committee On Radio Interference
EMI	electromagnetic interference
FTP	file transfer protocol
HotPnP	Hot Plug and Play
Hz	hertz
IEC	International Electrotechnical Commission
I/O	input/output
LAS	Link Active Scheduler
LED	Light-emitting diode
Link Active Scheduler	The Fieldbus device that is currently controlling access to the Fieldbus
OPC	OLE for Process Control
POST	power-on self test
RAM	random-access memory
V	Volts
VDC	Volts direct current

Index

Numbers

4-20 mA pressure sensor, converting to Fieldbus (example), 3-2 to 3-6
  assigning tag to new block, 3-3
  bringing block online, 3-6
  creating function block, 3-2 to 3-3
  scaling the reading, 3-4
  selecting module and channel, 3-3
  setting input range, 3-4
  setting up scheduling, 3-5

A

ACK_OPTION parameter, C-1
Actual modes (table), 4-11 to 4-12, C-14
address tag. *See* tags.
AI (Analog Input) function block
  connecting PID to AI and AO blocks (example), 3-14
  description, 4-2
  PID control loops, 4-5
alarm parameters, 4-6 to 4-8
  ALARM_STATE/UPDATE_STATE, 4-7
  SUBCODE subfield, 4-8
  TIME_STAMP, 4-8
  UNACKNOWLEDGED, 4-7
  VALUE, 4-8
ALARM_HYS parameter, C-1
alarming
  overview, 4-6
  temperature control with FP-3000 (example), 3-15 to 3-17
ALARM_STATE/UPDATE_STATE parameter, 4-7
ALARM_SUM parameter, C-1
ALERT_KEY parameter, C-1
ALG_RUN_TIME parameter, C-1
Analog Input function block. *See* AI (Analog Input) function block.
Analog Output function block. *See* AO (Analog Output) function block.
AO (Analog Output) function block
  connecting PID to AI and AO blocks (example), 3-14
  description, 4-2pacetopace 4-3
  PID control loops, 4-5
applications. *See* example applications.

B

BAL_TIME parameter, C-1
BINARY_CL parameter, C-1
BINARY_OP parameter, C-2
bitmasks for IO_OPTS parameter (table), C-11 to C-12
BKCAL_HYS parameter, C-2
BKCAL_IN parameter, C-2
BKCAL_OUT parameter, C-2
block instantiation
  overview, 1-3
  temperature control with FP-3000 (example)
    controlling heating element, 3-10
    PID control, 3-13
BLOCK_ALM parameter, C-2
BLOCK_ERR parameter (table), C-2 to C-4
BLOCK_RESET parameter (table), C-2 to C-4
blocks. *See also* function blocks.
  alarm parameters, 4-6 to 4-8
  alarming, 4-6

bringing online
converting 4-20 mA pressure sensor to Fieldbus (example), 3-6
temperature control with FP-3000 (example)
controlling heating element, 3-12
taking temperature readings, 3-10
FP-3000 specific parameters, 4-13 to 4-17
CFG_OPTS (table), 4-13
DEV_OPTS (table), 4-13
EXECUTION_STATISTICS (table), 4-14
FIELDPOINT_CHANNEL, 4-14
FIELDPOINT_MODULE, 4-14
FP_AI_100_RANGE, 4-14
FP_AI_110_RANGE, 4-14
FP_AI_111_RANGE, 4-15
FP_AO_200_RANGE, 4-15
FP_CJC_SOURCE, 4-15
FP_MOD_STATUS (table), 4-15
FP_NOISE_REJECTION, 4-16
FP_PWM_520_PERIOD, 4-16
FP_RTD_122_RANGE, 4-16
FP_RTD_TYPE, 4-16
FP_TC_120_CJ_RANGE, 4-16
FP_TC_120_RANGE, 4-16
FP_THERMOCOUPLE_TYPE, 4-16
LAST_BLOCK_EVENT (table), 4-16 to 4-17
VERSION_INFORMATION, 4-16 to 4-17
MODE_BLK parameter and mode handling, 4-10 to 4-12
Actual modes (table), 4-11 to 4-12
Normal mode, 4-12
Permitted mode, 4-12
Target modes (table), 4-10 to 4-11
overview, 4-1 to 4-2
PID control, 4-5 to 4-6
resource blocks, 4-2
status handling, 4-8 to 4-10
Limit subfield values (table), 4-10
Quality subfields (table), 4-9
Substatus field, 4-9
types of blocks, 4-1
BYPASS parameter, C-4

C

cascade initialization, D-1 to D-4
bypassing, D-4
mode and status behavior, D-2 to D-3
parameter connections, D-1 to D-2
remote cascades, D-3
CDO (Complete Discrete Output) function block
description, 4-3
interlock priorities (table), 4-3 to 4-4
CFG_OPTS parameter (table), 4-13
channel and module selection. *See* module and channel selection.
CHANNEL parameter, C-4
CHECKBACK states (table), C-5
CLF_FSTATE parameter, C-5
Complete Discrete Output (CDO) function block, 4-3 to 4-4
compliance specifications, E-1
configuration
configuration toggle switches (figure), A-1
HotPnP, 2-14 to 2-15
LED indicators, 2-12 to 2-14
Reset switch, A-2
Simulate Enable switch, A-1
software configuration problems, B-3 to B-6
generic software configuration problems (table), B-4 to B-6

overview, B-3
resource block configuration problems (table), B-6
updating FP-3000 firmware, 2-15 to 2-18
Write Lock switch, A-2
CONFIRM_TIME parameter, C-6
CONTROL_OPTS parameter (table), C-6
converting 4-20 mA pressure sensor to Fieldbus. *See* pressure sensor, converting to Fieldbus (example).
CYCLE_SEL/CYCLE_TYPE parameter, C-7

D

DD_RESOURCE parameter, C-7
DD_REV parameter, C-7
Device Description file (DD)
installing, 2-1 to 2-3
updating device description, 2-3
device tag. *See* tags.
DEV_OPTS parameter (table), 4-13
DEV_REV parameter, C-7
DEV_TYPE parameter, C-7
DI (Discrete Input) function block, 4-3
DIN rail, mounting FP-3000 on, 2-4 to 2-6
connecting terminal bases, 2-5 to 2-6
removing FP-3000, 2-6
Discrete Input (DI) function block, 4-3
Discrete Output (DO) function block, 4-3
DO (Discrete Output) function block, 4-3
documentation
conventions used in manual, *xvi*
organization of manual, *xv*
related documentation, *xvii*
DV_HI_ALM parameter, C-7
DV_HI_LIM parameter, C-7
DV_HI_PRI parameter, C-7
DV_LO_ALM parameter, C-7
DV_LO_LIM parameter, C-7
DV_LO_PRI parameter, C-7

E

electronic support services, F-1
e-mail support, F-1
environment specifications, E-1
error codes for BLOCK_ERR parameter (table), C-2 to C-4
example applications, 3-1 to 3-17
converting 4-20 mA pressure sensor to Fieldbus, 3-2 to 3-6
assigning tag to new block, 3-3
bringing block online, 3-6
creating function block, 3-2 to 3-3
scaling the reading, 3-4
selecting module and channel, 3-3
setting input range, 3-4
setting up scheduling, 3-5
initial power on: assigning address and device tag, 3-1
temperature control with FP-3000, 3-6 to 3-17
alarming, 3-15 to 3-17
controlling heating element, 3-10 to 3-12
getting started, 3-6 to 3-7
PID control, 3-13 to 3-14
taking temperature readings, 3-7 to 3-10
EXECUTION_STATISTICS parameter (table), 4-14

F

fault state for function blocks, D-4
FAULT_STATE parameter, C-8
fax and telephone support numbers, F-2
Fax-on-Demand support, F-1
FEATURE_SEL_FEATURES parameter (table), C-8
FF_GAIN parameter, C-9
FF_SCALE parameter, C-9
FF_VAL parameter, C-9

field upgradability, 1-4
Fieldbus communication problems
overview, B-1
problems and solutions (table), B-2
setting device tag and network address, B-1
Fieldbus network, connecting to FP-3000, 2-11 to 2-12
Fieldbus parameters, C-1 to C-23
ACK_OPTION, C-1
ALARM_HYS, C-1
ALARM_SUM, C-1
ALERT_KEY, C-1
ALG_RUN_TIME, C-1
BAL_TIME, C-1
BINARY_CL, C-1
BINARY_OP, C-2
BKCAL_HYS, C-2
BKCAL_IN, C-2
BKCAL_OUT, C-2
BLOCK_ALM, C-2
BLOCK_ERR (table), C-2 to C-4
BLOCK_RESET (table), C-2 to C-4
BYPASS, C-4
CHANNEL, C-4
CHECKBACK states (table), C-5
CLF_FSTATE, C-5
CONFIRM_TIME, C-6
CONTROL_OPTS (table), C-6
CYCLE_SEL/CYCLE_TYPE, C-7
DD_RESOURCE, C-7
DD_REV, C-7
DEV_REV, C-7
DEV_TYPE, C-7
DV_HI_ALM, C-7
DV_HI_LIM, C-7
DV_HI_PRI, C-7
DV_LO_ALM, C-7
DV_LO_LIM, C-7
DV_LO_PRI, C-7
FAULT_STATE, C-8
FEATURE_SEL_FEATURES (table), C-8
FF_GAIN, C-9
FF_SCALE, C-9
FF_VAL, C-9
FIELD_VAL, C-9
FP_AUTOCONFIGURE, C-9
FP_MOD_LIST, C-9
FREE_SPACE, C-9
FREE_TIME, C-9
GAIN, C-10
GRANT_DENY, C-10
HI_ALM, C-10
HI_HI_ALM, C-10
HI_HI_LIM, C-10
HI_HI_PRI, C-10
HI_LIM, C-11
HI_PRI, C-11
IO_OPTS (table), C-11 to C-12
LIM_NOTIFY, C-12
LO_ALM, C-12
LO_LIM, C-13
LO_LO_ALM, C-13
LO_LO_LIM, C-13
LO_LO_PRI, C-13
LO_PRI, C-13
LOW_CUT, C-13
L_TYPES (table), C-12
MANUFAC_ID, C-13
MAX_NOTIFY, C-13
MEMORY_SIZE, C-13
MIN_CYCLE_T, C-13
MODE_BLK (table), C-14
NV_CYCLE_T, C-15
OP_CMD_CXO (table), C-15
OUT, C-15
OUT_HI_LIM, C-15
OUT_LO_LIM, C-16
OUT_SCALE (table), C-16
PV, C-16
PV_FTIME, C-16

PV_SCALE, C-16
RATE, C-16
RCAS_IN, C-17
RCAS_OUT, C-17
RESET, C-17
RESTART (table), C-17
ROUT_IN, C-17 to C-18
ROUT_OUT, C-18
RS_STATE (table), C-18
SAFEGUARD_CL, C-18
SAFEGUARD_OP, C-18
SET_FSTATE, C-19
SHED_OPT (table), C-19
SHED_RCAS, C-19
SHED_ROUT, C-19
SIMULATE, C-20
SP_HI_LIM, C-20
SP_LO_LIM, C-20
SP_RATE_DN, C-20
SP_RATE_UP, C-20
STATUS_OPTS (table), C-21
STRATEGY, C-22
ST_REV, C-20
TAG_DESC, C-22
TEST_RW, C-22
TRK_IN_D, C-22
TRK_SCALE, C-22
TRK_VAL, C-22
UPDATE_EVT, C-22
WRITE_ALM, C-22
WRITE_LOCK, C-22
WRITE_PRI, C-22
XD_SCALE (table), C-23
FIELDPOINT_CHANNEL parameter, 4-14
FIELDPOINT_MODULE parameter, 4-14
FIELD_VAL parameter, C-9
firmware for FP-3000
field upgradability, 1-4
updating, 2-15 to 2-18
FP-3000 connector pinout (figure), 2-12
FP-3000 Network Module
features, 1-3 to 1-4
overview, 1-1 to 1-2
setup (figure), 1-2
FP_AI_100_RANGE parameter, 4-14
FP_AI_110_RANGE parameter, 4-14
FP_AI_111_RANGE parameter, 4-15
FP-AO-200 block, instantiating, 3-10
FP_AO_200_RANGE parameter, 4-15
FP_AUTOCONFIGURE parameter, C-9
FP_CJC_SOURCE parameter, 4-15
FP_MOD_LIST parameter, C-9
FP_MOD_STATUS parameter (table), 4-15
FP_NOISE_REJECTION parameter, 4-16
FP_PWM_520_PERIOD parameter, 4-16
FP_RTD_122_RANGE parameter, 4-16
FP_RTD_TYPE parameter, 4-16
FP-TC-120 block, creating, 3-7
FP_TC_120_CJ_RANGE parameter, 4-16
FP_TC_120_RANGE parameter, 4-16
FP_THERMOCOUPLE_TYPE parameter, 4-16
FREE_SPACE parameter, C-9
FREE_TIME parameter, C-9
FTP support, F-1
function blocks. *See also* blocks.
advanced features, D-1 to D-5
blocks used in PID control loops, 4-2
cascade initialization, D-1 to D-4
bypassing, D-4
mode and status behavior, D-2 to D-3
parameter connections, D-1 to D-2
remote cascades, D-3
converting 4-20 mA pressure sensor to Fieldbus (example), 3-2 to 3-3
fault state, D-4
mode shedding, D-4 to D-5
overview, 1-3
purpose and use, 4-1 to 4-2

support for FieldPoint modules, 4-4 to 4-5
types of function blocks, 4-2 to 4-4

G

GAIN parameter, C-10
GRANT_DENY parameter, C-10

H

heating element, controlling (example), 3-10 to 3-12
assigning tag to new block, 3-10
bringing block online, 3-12
instantiating FP-AO-200 block, 3-10
scaling output, 3-11 to 3-12
selecting module and channel, 3-11
setting output range, 3-11
setting up scheduling, 3-12
HI_ALM parameter, C-10
HI_HI_ALM parameter, C-10
HI_HI_LIM parameter, C-10
HI_HI_PRI parameter, C-10
HI_LIM parameter, C-11
HI_PRI parameter, C-11
HotPnP
avoiding damage to network module and terminal bases (note), 2-14
inserting new I/O modules, 2-15
overview, 1-4
replacing I/O modules, 2-15
using during operation, 2-14 to 2-15

I

initial power on procedure, 3-1
input range, setting
converting 4-20 mA pressure sensor to Fieldbus (example), 3-4
temperature control with FP-3000 (example), 3-8
installation
connecting FP-3000 to Fieldbus network, 2-11 to 2-12
connecting power to FP-3000, 2-9 to 2-10
Device Description file, 2-1 to 2-3
HotPnP, 2-14 to 2-15
LED indicators, 2-12 to 2-14
mounting FP-3000 on DIN rail, 2-4 to 2-6
connecting terminal bases, 2-5 to 2-6
removing FP-3000, 2-6
mounting FP-3000 to a panel, 2-6 to 2-8
connecting terminal bases, 2-7 to 2-8
removing FP-3000, 2-8
mounting I/O modules onto terminal bases, 2-8
power-on-self test (POST), 2-10
updating FP-3000 firmware, 2-15 to 2-18
instantiating blocks. *See* block instantiation.
interoperability of FP-3000, 1-4
I/O modules
installing with HotPnP, 2-14 to 2-15
inserting new I/O modules, 2-15
replacing I/O modules, 2-15
problems (table), B-3
IO_OPTS parameter (table), C-11 to C-12

L

L_TYPES parameter (table), C-12
LAS (Link Active Scheduler), 1-4
LAST_BLOCK_EVENT parameter (table), 4-16 to 4-17
LED indicators, 2-12 to 2-14
description of LED states (table), 2-13
STATUS LED flashes and error conditions (table), 2-14
Limit subfield values for status handling (table), 4-10
LIM_NOTIFY parameter, C-12

linearization types (table), C-12
Link Active Scheduler (LAS), 1-4
LO_ALM parameter, C-12
LO_LIM parameter, C-13
LO_LO_ALM parameter, C-13
LO_LO_LIM parameter, C-13
LO_LO_PRI parameter, C-13
LO_PRI parameter, C-13
LOG (FieldPoint Log Block) function block, 4-4
LOW_CUT parameter, C-13

M

manual. *See* documentation.
MANUFAC_ID parameter, C-13
MAX_NOTIFY parameter, C-13
MEMORY_SIZE parameter, C-13
MIN_CYCLE_T parameter, C-13
mode and status behavior during cascade initialization, D-2 to D-3
mode shedding for function blocks, D-4 to D-5
  options (table), D-5
MODE_BLK parameter and mode handling, 4-10 to 4-12, C-14 to C-15
  Actual modes (table), 4-11 to 4-12, C-14
  Normal mode, 4-12, C-15
  Permitted mode, 4-12, C-15
  Target modes (table), 4-10 to 4-11, C-14
module and channel selection
  converting 4-20 mA pressure sensor to Fieldbus, 3-3
  temperature control with FP-3000 (example)
    controlling heating element, 3-11
    taking temperature readings, 3-7
mounting FP-3000. *See* installation.

N

network specifications, E-1
Normal mode, 4-12, C-16
NV_CYCLE_T parameter, C-15

O

OP_CMD_CXO parameter (table), C-15
OUT parameter, C-15
OUT_HI_LIM parameter, C-15
OUT_LO_LIM parameter, C-16
OUT_SCALE parameter (table), C-16
output
  scaling (example), 3-11 to 3-12
  setting range (example), 3-11

P

Permitted mode, 4-12, C-15
PID control, 4-5 to 4-6
  function blocks in PID control loop, 4-5
  overview, 1-3
  PID Function Block Application dialog box (figure), 4-6
  temperature control with FP-3000 (example), 3-13 to 3-14
    assigning tag to new block, 3-13
    connecting PID to AI and AO blocks, 3-14
    downloading and bringing loop into Auto, 3-14 to 3-15
    instantiating PID block, 3-13
    scaling the PID, 3-13
    tuning the PID, 3-15
PID (Proportional-Integral-Derivative) function block
  description, 4-3
  PID control loops, 4-5
Plug and Play. *See* HotPnP.
POST (power-on self test), 2-10

power
calculating power for FieldPoint bank, 2-10
connecting power to FP-3000, 2-9
initial power on procedure, 3-1
POWER LED, 2-12
power-on self test (POST), 2-10
4-20 mA pressure sensor, converting to Fieldbus (example), 3-2 to 3-6
assigning tag to new block, 3-3
bringing block online, 3-6
creating function block, 3-2 to 3-3
scaling the reading, 3-4
selecting module and channel, 3-3
setting input range, 3-4
setting up scheduling, 3-5
problem solving. *See* troubleshooting.
PROCESS LED, 2-13
Proportional-Integral-Derivative function block. *See* PID control; PID (Proportional-Integral-Derivative) function block.
PV parameter, C-16
PV_FTIME parameter, C-16
PV_SCALE parameter, C-16

Q

Quality subfields for status handling (table), 4-9
questions and answers. *See* troubleshooting.

R

RATE parameter, C-16
RCAS_IN parameter, C-17
RCAS_OUT parameter, C-17
READY LED, 2-12
remote cascades, D-3
RESET parameter, C-17
Reset switch, A-2
resource block
configuration problems (table), B-6
description, 4-2
RESTART parameter (table), C-17
ROUT_IN parameter, C-17 to C-18
ROUT_OUT parameter, C-18
RS_STATE parameter (table), C-18

S

SAFEGUARD_CL parameter, C-18
SAFEGUARD_OP parameter, C-18
scaling the reading
converting 4-20 mA pressure sensor to Fieldbus (example), 3-4
temperature control with FP-3000 (example), 3-8 to 3-9
scheduling, setting up
converting 4-20 mA pressure sensor to Fieldbus (example), 3-5
temperature control with FP-3000 (example)
controlling heating element, 3-12
taking temperature readings, 3-9
SET_FSTATE parameter, C-19
SHED_OPT parameter (table), C-19
SHED_RCAS parameter, C-19
SHED_ROUT parameter, C-19
Simulate Enable switch, A-1
SIMULATE parameter, C-20
software configuration problems, B-3 to B-6
generic software configuration problems (table), B-4 to B-6
overview, B-3
resource block configuration problems (table), B-6
specifications
compliance, E-1
environment, E-1
network, E-1
SP_HI_LIM parameter, C-20

SP_LO_LIM parameter, C-20
SP_RATE_DN parameter, C-20
SP_RATE_UP parameter, C-20
STAT (FieldPoint Statistics Block) function block, 4-4
status and mode behavior during cascade initialization, D-2 to D-3
status handling, 4-8 to 4-10
  Limit subfield values (table), 4-10
  Quality subfields (table), 4-9
  Substatus field, 4-9
STATUS LED
  flashes and error conditions (table), 2-14
  operation, 2-13
STATUS_OPTS parameter (table), C-21
STRATEGY parameter, C-22
ST_REV parameter, C-20
Substatus field, 4-9

T

TAG_DESC parameter, C-22
tags
  assigning address and device tag at initial power on, 3-1
  assigning tag to new block
    converting pressure sensor to Fieldbus (example), 3-3
    temperature control with FP-3000 (example)
      controlling heating element, 3-10
      PID control, 3-13
      taking temperature readings, 3-7
  setting device tag and network address, B-1
Target modes (table), 4-10 to 4-11, C-14
technical support, F-1 to F-2. *See also* troubleshooting.
telephone and fax support numbers, F-2
temperature control with FP-3000, 3-6 to 3-17
  alarming, 3-15 to 3-17
  controlling heating element, 3-10 to 3-12
    assigning tag to new block, 3-10
    bringing block online, 3-12
    instantiating FP-AO-200 block, 3-10
    scaling output, 3-11 to 3-12
    selecting module and channel, 3-11
    setting output range, 3-11
    setting up scheduling, 3-12
  getting started, 3-6 to 3-7
  PID control, 3-13 to 3-14
    assigning tag to new block, 3-13
    connecting PID to AI and AO blocks, 3-14
    downloading and bringing loop into Auto, 3-14 to 3-15
    instantiating PID block, 3-13
    scaling the PID, 3-13
    tuning the PID, 3-15
  taking temperature readings, 3-7 to 3-10
    assigning tag to new block, 3-7
    bringing block online, 3-10
    creating FP-TC-120 block, 3-7
    scaling the reading, 3-8 to 3-9
    selecting module and channel, 3-7
    setting input range and thermocouple type, 3-8
    setting up scheduling, 3-9
terminal bases
  connecting
    with DIN rail mounting, 2-5 to 2-6
    with panel mounting, 2-7 to 2-8
  mounting I/O modules onto, 2-8
TEST_RW parameter, C-22
TIME_STAMP alarm parameter, 4-8
TRK_IN_D parameter, C-22
TRK_SCALE parameter, C-22
TRK_VAL parameter, C-22
troubleshooting, B-1 to B-6
  Fieldbus communication problems overview, B-1

problems and solutions (table), B-2
setting device tag and network address, B-1
I/O module problems (table), B-3
software configuration problems, B-3 to B-6
generic software configuration problems (table), B-4 to B-6
overview, B-3
resource block configuration problems (table), B-6

U

UNACKNOWLEDGED alarm parameter, 4-7
UPDATE_EVT parameter, C-22
updating firmware for FP-3000, 2-15 to 2-18

V

VALUE alarm parameter, 4-8
VERSION_INFORMATION parameter, 4-16 to 4-17

W

Write Lock switch, A-2
WRITE_ALM parameter, C-22
WRITE_LOCK parameter, C-22
WRITE_PRI parameter, C-22

X

XD_SCALE parameter (table), C-23